With This Issue
Classified Section - p98
Software Security Supplement.

.EXE

The Software Developers' Magazi

FEBRUARY 1992 VOL 6 ISSUE 8

Don't judge us by our cover -
This issue we look at Clipper.

Where do we go from five-oh?
The Clipper language grows up.

How to persist TBrowse objects.

There are four new regular columns,
To please C lovers and C haters.

Five Programmer's Editors compared.

And, oh yes, there's Microsoft's C++.
An exclusive try-out of a Beta called Pete.

Introducing the First Complete Education Programme in Object-Oriented Technology

OBJECT TECHNOLOGY STRATEGIC BRIEFING

OBJECT-ORIENTED PRIMER

If properly implemented the benefits of object technology to software developers can be immense, but as an emergent technology there are as many pitfalls as opportunities.

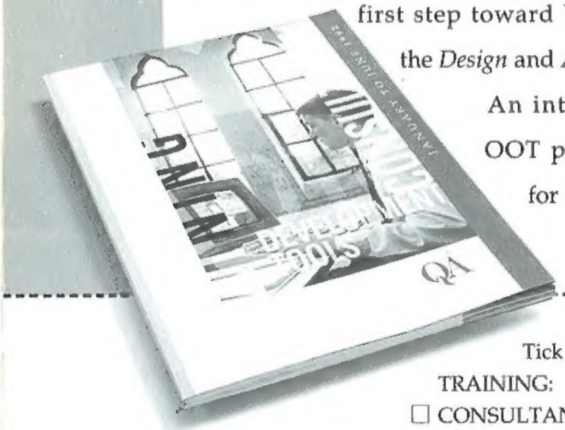
QA has a unique and unified education programme in Object-Oriented Technology (OOT) which will help you address the issues and deliver the benefits.

The cornerstone of our programme is the management-level *Strategic Briefing* explaining the benefits of OOT and how to evaluate, pilot and implement projects. An essential to all those who will be involved in OOT decision making.

Alternatively, you may prefer to begin with the *Primer* course, a technical introduction to OOT and an excellent first step toward both the *Design* and *Analysis* courses.

CONSULTANCY

An integral part of our OOT programme is that for C++, one of the industry's most popular implementation languages to achieve your Object-Oriented goals.



Expert Consultation

OBJECT-ORIENTED ANALYSIS

OBJECT-ORIENTED DESIGN

C++ PROGRAMMING

As a result of our total commitment to OOT and the considerable expertise we have now developed in the subject, your company may prefer to explore and develop its commercial opportunities through our consultancy service.

You are invited to start your own chosen path to OOT mastery today:

To obtain full details of our carefully structured courses and consultancy in this new technology, which is set to revolutionise the world of software development, simply telephone Samantha Trinder on 0285 655888 or return the coupon.

Mail to QA Training Ltd, Cecily Hill Castle, Cirencester, Gloucestershire GL7 2EF, UK.

Tick here for details:

TRAINING: □ Object-Oriented □ Unix/Aix □ Networks □ OS/2 □ Windows □ C

□ CONSULTANCY

□ SOFTWARE DEVELOPMENT PRODUCTS

NAME ____________ TITLE ____________

.EXE 2/92

COMPANY ____________

ADDRESS ____________

TEL. NO. ____________

AEXE1

QA TRAINING

First for Object-Oriented Education

CIRCLE NO. 426

Vol 6 Issue 8

.EXE

February 1992

Editor: Willie Watts
Features Editor: Paul Kemp
Staff Writer: Cliff Saran
Production Manager: Katerina Adams
Design & Layout: Mark English
Group Advertising Manager: Sandra Inniss-Palmer
Assistant Advertising Manager: Ed Butcher
Advertising Executive: Marc Warren
Reader Services & Promotions: Helena Adams
Administration & Subscriptions: Rena Gibbs, Sandy McDonnell
Front Cover Photography: David Pritchard

General

.EXE Magazine is independent and not affiliated to any vendor of hardware, software or services. It is published by Process Communications Ltd, 10 Barley Mow Passage, Chiswick, London W4 4PH.
Tel: (Advertising/Editorial/Production) 081 994 6477
(Subscriptions) 0442 824501
Facsimile: 081 994 1533 Telex: 8811418 SPACES G
ISSN: 0268-6872

Subscriptions

.EXE Magazine is a monthly journal for software developers. It is available only by subscription, at a cost of £35 per annum (11 issues) in the UK. 'A subscription implies that this journal will be sent to the subscriber until one of the three expires' - AG Macdonell. The magazine is published around the 26th of the month preceding the cover date. There is no January issue. To subscribe or obtain details of overseas/academic rates, please call 044282 4501, or write to The Subscriptions Manager, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We can invoice your company or take ACCESS and VISA cards. In case of a query about your subscription, please call 044282 4501.
Back issues are available at a cost of £3.50 each - call our Chiswick office (081 994 6477) for a list of issues still in print.

Editorial

Editorial enquiries should be addressed to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. We welcome letters, opinions, suggestions and articles from our readers. Please write for a copy of our Contributors' Guide.
Information contained in .EXE is believed to be correct. If errors are found, we will endeavour to publish a clarification in the next available issue.
From time to time, we offer to copy the PC program code described in an article onto diskette for our readers. In this case, please send a blank, formatted disk with a self-addressed, prepaid mailer to the editorial address given above. We can copy both 5.25" and 3.5" disks.
The publishers can accept no liability for any consequences of using software distributed in this way.

Advertising

If you are interested in advertising in this magazine, please write to the Display Advertising Manager/Recruitment Sales Manager (as appropriate), .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH, or call 081 994 6477 for details of our advertising rates.

Pronunciation

The name of .EXE Magazine is pronounced to rhyme with 'not sexy magazine'.

Copyright

Issue theme: Clipper

The metaphor that went wrong

John Barber isn't happy with the way computers treat files.

When you use metaphors that are familiar and real-world based, users can transfer previous knowledge of their work environment to your application interface. You must be careful in choosing a metaphor to make sure the metaphor meets the expectations users have because of their real world experience. - Common User Access Advanced Interface Design Guide.

If you were asked to think of an example where this advice had been successfully applied, what would you choose? The first thing which comes to my mind is the spray can of a Paint program. The user's real world knowledge can be transferred in such an obvious way that it must be very unusual for anyone to reach for the manual. You just pick up the can and press the button.

Unfortunately, not all metaphors are this successful. One in particular has suffered from inconsistency and poor implementation. It's been around a lot longer than the spray can, and it's so widely used that it's easy to forget it is a metaphor. It is ...FILE.

I don't know who first used the word in connection with computers, but no doubt they were likening computer storage to the traditional office filing cabinet. Yet it seems that this potentially helpful metaphor has been obscured by later developments. In modern GUI application programs, a standard format File Menu has emerged, in conformance with CUA guidelines, but even here, close scrutiny reveals some serious flaws.

First, the menu seems to be designed to keep the user in the dark about what is really happening. One important concept which all users need to be aware of is that programs operate on a working copy of data, not the original. There is nothing difficult about this. If we were modifying a document manually using a pen, it would be sensible to take a photocopy and work on that, leaving the original document safely in its file until we were happy with the modification. Programs do the same. Yet many inexperienced users must be fooled by commands such as File Open and File Print, and believe that they are working directly on the contents of the file. For them, opening up a file is rather like a surgeon opening up a patient.

This difficulty could be fixed quite easily by some rewording of the menu items. A further problem becomes apparent if we take the manual editing analogy a little further. After we have created the modified document, we can file it and destroy the original. But sometimes, we may choose to keep the original also, at least for a while. This too is a common event in our real world experience. The equivalent action on the standard file menu, 'File Save As ...', is hardly intuitive, and the user has the inconvenience of having to think up a new name for what may be just a new version of the same thing.

Having multiple versions of documents is such a natural idea in our real world experience that it should be an intrinsic part of any computer filing system worth its salt. The concept of version control is just as relevant to the users of application programs as to the software developers who create the programs.

Why should users, who may be non-technical, be expected to navigate around a tree-structured directory system? Once we introduce tree-structured filing systems, 'File' as a metaphor begins to loose its consistency. Filing cabinets do not grow on trees.

Obviously, files need to be organised in some way. The important thing is the interface which is presented to the user. To locate a file, it should simply be a matter of stating what we know about it, eg that it is a letter and that it was written by Jim. The problem with path names is that \JIM\LETTERS and \LETTERS\JIM do not mean the same thing. The ordering of the information which occurs in a path name is quite unnecessary for the user's purposes.

It seems to me that while many aspects of software usability have improved dramatically, those related to files have been sadly neglected. At the heart of the problem is the implementation of a metaphor in a way which doesn't meet anyone's expectations.

EXE

John Barber was once a respectable mechanical engineer, but his reputation was ruined by a brief but passionate affair with a Hewlett Packard 9825A in the early 1980's. Since then he has indulged in many programming excesses and is now reduced to a life of toil and poverty as a freelance programmer.

GREY MATTER

Prigg Meadow, Ashburton, Devon TQ13 7DF

Prices do not include VAT or other local taxes, but do include delivery in the UK and Europe. Please check prices at time of order as ads are prepared some weeks before publication. This page lists some of our products– call us for a complete price list and details of discount structure. ORDER BY PHONE WITH YOUR CREDIT CARD.

TEL: (0364) 53499

FAX: (0364) 53071

PROGRAM EDITORS

NEW MULTI-EDIT V6 includes new C-like macro language, real-time DOS output window, faster virtual memory manager, binary & Unix file editing, XMS support, etc, etc. CALL US FOR MORE DETAILS, including update pricing.

NEW MULTI-EDIT LITE must be the bargain editor of the year! No macro language, but most other features of Standard. ONLY £49.

OTHER NEW MULTI-EDIT ADD-ONS: MULTI-TAGS hypertext source code browser @ £29, EVOLVE for dBase or Clipper @ £65, GUIDE.2 context-sensitive on-line hypertext reference manual for C++ @ £35 or Turbo Pascal 6 @ £20.

Brief	DOS	£199
EC Editor	DOS	£40
Epsilon v5	DOS	£165
Kedit v4	DOS	£95
Multi-Edit Lite v6	DOS	£49
Multi-Edit Standard v6	DOS	£75
Multi-Edit Professional v6	DOS	£110
Norton Editor v2	DOS	£45
PVCS Prof Editor	DOS	£195
QEdit Advanced	DOS	£45
Vedit-Plus v3.5	DOS	£120
Kedit v4	OS2&DOS	£105
Epsilon v5 for OS/2	OS2	£165
QEdit Advanced for OS/2	OS2	£62
PVCS Prof Editor	OS2	£195
MIX Editor	CPM-Z80	£25
Vedit-Plus v2.33	CPM-80	£110

For more information please call us.

PROGRAMMING TOOLS

Ada Compilers
Assemblers & Libs
Basic Compilers
Basic Utilities
C Compilers
C Interpreters
C++
Comms.Libraries
Database Libs.
Dis-assemblers
Forth
Fortran Libraries
Icon
Lisp
Modula-2
Pascal Compilers
Prolog
Screen Libraries

Algol Compilers
AWK
Basic Interpreters
Basic Libraries
C Utilities
C Libraries
Cobol Compilers
Cross Assemblers
Debuggers
Editors
Fortran Compilers
Graphics Libraries
Linkers/Locaters
Logo
Object Oriented
Pascal Libraries
Rexx
Smalltalk

We stock many items for which there is no space in these advertisements.

PROLOG LANGUAGE

Arity Standard Prolog	DOS	£82
Arity/Prolog Comp. v6	DOS	£515
Arity/Prolog Int. v6	DOS	£220
Educ.Prolog (ADA)	DOS	£28
FS Prolog (ADA)	DOS	£45
VMI Prolog (ADA)	DOS	£75
VML Prolog (ADA)	DOS	£135
LPA Prolog Prof Prog Ed	DOS	£300
Flex Pack Prog Edition	DOS	£750
micro-Prolog v3.1	DOS	£69
PDC Prolog	DOS	£240
Prolog++ Pack Prog Ed	DOS	£375
Arity/Prolog Comp	OS2	£745
Micro-Prolog v3.1	CPM-86	£69
Micro-Prolog v3.1	CPM-Z80	£69
LPA Mac Prolog v3	MAC	£450

C LIBRARIES & TOOLS

Comms

C Asynch Manager	DOS	£130
C Comms Toolkit	DOS	£89
Essential Comms	DOS	£180
Greenleaf CommLib 3.2	DOS	£195
Silvercomm C Asynch	DOS	£140

Database

Btrieve	OS2,Win,DOS	£390
C/Database Toolchest	DOS	£45
C-Index/Plus	DOS	£175
C-tree Plus	DOS	£350
CodeBase 4.5	DOS	£215
CQL	DOS	£270
D-ISAM	DOS	£325
db-VISTA DBMS Object	DOS	£425
Essential Btree	DOS	£110
Faircom Toolbox Prof v2	DOS	£680
Pro-C & Workbench	DOS	£799
SoftFocus Btree/ISAM	DOS	£80
Sycero.C Network	DOS	£895

Graphics

Essential Graphics Kernel	DOS	£130
Essential Graphics GUI	DOS	£105
GFX Graphics	DOS	£80
GraphiC 6.0	DOS	£305
graphics-MENU & Data Entry	DOS	£180
GSS GDT	OS2,DOS	£425
GX Graphics	DOS	£120
GX Effects	DOS	£120
GX Text	DOS	£90
HALO Professional 2.0	DOS	£270
HGraph	DOS	£80
MetaWINDOW	DOS	£165
MetaWINDOW/Plus	DOS	£220
PCX Programmer's Toolkit	DOS	£150
Real-Time Graphics & M/CT	DOS	£145
TEGL Windows Toolkit II	DOS	£70

Screen

C Windows Toolkit	DOS	£59
C-Scape + Look&Feel	OS2&DOS	£370
C-Worthy & cwArchitect	DOS	£389
Greenleaf Datawindows	DOS	£215
MEWEL Window System	DOS	£195
MEWEL Toolbox	DOS	£165
Panel Plus II	DOS	£260
ProtoView	Win	£750
TesSeRact CXL	DOS	£42
Vermont Views V3.0	DOS	£325
Vitamin C V4.0	DOS	£250

General & Sundry Libraries

C Tools Plus/6.0	DOS	£105
CodeRunner	DOS	£140
C Utility Library	DOS	£140
Greenleaf Functions	DOS	£125
Heap Checker	DOS	£30
Heap Expander 3.0	DOS	£85
Hold Everything	DOS	£130
PDK £1, £2, £3	DOS	£70
PowerSTOR	DOS	£85
Resident-C	DOS	£135
TesSeRact TSR Dev Sys	DOS	£42
Turbo C Tools/2.0	DOS	£105
WKS Library	DOS	£160

Math & Scientific

C/Math Toolchest & Grafix	DOS	£45
C Lang Sci Lib	ANY C	£160
Huge Virtual Array & NAT	DOS	£215
Mathpak 87	DOS	£85
Science & Eng Tools	DOS	£79

Tools

C/Utilities Toolchest	DOS	£45
C-Clearly	DOS	£85
Clear+ for C	DOS	£140
CodeCheck Prof	DOS	£435
LALR Pro 4.2	DOS	£275
MKS LEX & YACC	DOS	£165
PC-Lint	OS2&DOS	£85
PCYACC Prof	DOS	£435
Source Print	DOS	£75
Tree Diagrammer	DOS	£75

ASSEMBLERS

MS MASM PDS v6	OS2&DOS	£75
PharLap 386IDOS Ext SDK	DOS	£330
PharLap 386IASM/Linkloc	DOS	£830
Optasm	DOS	£125
Turbo Debugger & Tools	DOS	£89
2500AD Z80 ASM	CPM-Z80	£70
SLR Z80ASM	CPM-Z80	£40
SLR Z80ASM-PLUS	CPM-Z80	£140
SLR MAC	CPM-Z80	£40
SLR MAC-PLUS	CPM-Z80	£140
SLR 180 (64180)	CPM-Z80	£40
SLR 180-PLUS (64180)	CPM-Z80	£140

THE C LANGUAGE

Refer also to the C++ section.

C COMPILERS

NEW MICROSOFT C/C++ 7.0 - see C++ section. Only £150 for existing Microsoft C owners (if bought before 1/2/92 - we need proof of purchase).

Aztec C86-c	DOS	£275
Microsoft C 6.0	OS2&Win&DOS	£222
MS C & Win SDK	OS2&Win&DOS	£335
MS QuickC 2.5	DOS	£50
MS QuickC & QuickAsm	DOS	£83
MS QuickC for Windows	Win	£95
Topspeed C Std	DOS	£110
Topspeed C Prof	Win&DOS	£165
Watcom C 8.5	OS2&Win&DOS	£295
High C 386 Local 1.7	PL386	£585
High C 386 Global 2.3	PL386	£650
Intel 386 C Code Builder	Win&386	£399
NDP 386 C	386	£545
Watcom C 8.5/386	Win&386	£495
Topspeed C Prof	OS2	£165
Aztec CII-p	CPM-80	£120
Aztec CII-c	CPM-80	£200
Hi-Tech C	CPM-Z80	£105
Mix C	CPM-Z80	£35
Aztec C68K MPW C	MAC	£110
Aztec C68K/Am-P	AMIGA	£120
Aztec C68K/ST-P	ATARI	£120
Hi-Tech C 68000	ATARI	£140
Prospero C	ATARI	£55

We supply 2500AD, Avocet, Aztec, Lattice, IAR and Hi-Tech Cross Compilers hosted on DOS and targeted on Z80, 6502, 6801, 68HC11, 6301, 6809, 7811, 8051, 8096 & 68000/020. Please call for information or advice.

C INTERPRETERS

New V5 Instant-C is MS C V6 compatible.

C-terp V3.5	DOS	£165
Instant C V5	DOS	CALL

LOW PRICES FOR MICROSOFT & BORLAND PRODUCTS

Microsoft Basic PDS 7.1	**£222**
Microsoft C/C++ 7.0	**£222**
Microsoft C & Windows SDK	**£335**
Microsoft Cobol 4.5	**£435**
Microsoft Fortran 5.1	**£205**
Microsoft Macro Assembler 6.0	**£75**
Microsoft Pascal 4.0	**£139**
Microsoft QuickBASIC 4.5	**£50**
Microsoft QuickC 2.51	**£50**
Microsoft QuickC & Assembler	**£83**
Microsoft QuickC for Windows	**£95**
Microsoft QuickPascal 1.0	**£50**
Microsoft Source Profiler	**£45**
Microsoft Visual Basic	**£95**
Microsoft Windows 3.0	**£70**
Microsoft Windows SDK 3.0	**£225**
MS Multimedia Dev Kit	**£225**
MS MDK & SB Pro Card	**£365**
Borland C++ 3.0	**£199**
Borland C++ & AFX 3.0	**£295**
Borland Corporate Prod Pack	**£330**
Turbo C++	**£50**
Turbo C++ & Turbo Vision	**£69**
Turbo C++ for Windows	**£85**
Turbo Debugger & Tools	**£89**
Turbo Pascal 6.0	**£69**
Turbo Pascal Professional 6.0	**£137**
Turbo Pascal for Windows	**£103**
Paradox 3.5	**£340**
Paradox Engine 2.0	**£223**

WITH FULL TECHNICAL SUPPORT

DISK COPYING SERVICE

We can copy files to and from 600 disk formats including CP/M, CP/M-86, MS- DOS, PC-DOS, APPLE, SIRIUS, BBC, TORCH, APRICOT, HP-150, TRSDOS, AMSTRAD, ATARIST, MACINTOSH. Our charge is £10.00 + disk + VAT with discounts on small quantities and disks are normally despatched within 24hrs of receipt. For more information call us.

CIRCLE NO. 427

News

Village People

Don't forget the Windows Show 1992, billed as the UK's largest ever software event, which is to be staged at Olympia 2, London from 18th-20th February. A central attraction will doubtless be the .EXE Development Tools Village; the big names in Windows development will all be represented, including Microsoft, Borland and Symantech. For more information call IT Events Ltd on 0256 381456.

Eiffel comes to DOS

Would you like to be into OOP, but don't fancy something that's got C in it? German company SiG GmbH has released Eiffel/S, a DOS implementation of V3.0 of the language. The Eiffel/S compiler outputs C code - so you'll still need a C compiler - and comes with a set of basic class libraries which include classes for I/O, persistence and data structures. Prices start under £400, more info from UK distributor Applied Logic Developments (081 7801088).

Blinker outgrows Clipper

Blink Inc has begun shipping Blinker V2.0, the latest release of the popular dynamic overlay linker. This now integrates a memory swapper, using EMS/XMS memory to enable an executing program to generate a DOS shell for running a second program. There is also support for several languages including Microsoft C, Watcom C, Zortech C++, Borland C++, Clipper V5.01 and Assembler. Blinker costs £189 and is distributed in the UK by QBS on 081 9944842.

Win £20,000

There's a chance to win £20,000 in the Business Software Challenge '90. To be eligible you must be a student at a UK university, polytechnic or other education establishment during the academic year 1991-1992 (or you were a student in 1990-1991). All you have to do is to produce prototype software that identifies an area in the business world which would benefit from being computerised. Entries may be submitted by individuals or by teams of developers. The closing date for the competition is 30th April 1992. For details, write to The Competition Secretary, Business Software Challenge '90, 24-28 Bloomsbury Way, London WC1A 2PX. You can also telephone on 071 8316262.

VB to SQL Server

Microsoft has shipped its Visual Basic Library and SDK for SQL Server. The RRP is £335, Microsoft is on 0734 500741.

Clarion buys out JPI

Clarion Software and UK-based Jensen & Partners International (JPI) have announced an intent to merge the two companies. The merger will be accomplished by issuing Clarion stock to JPI stockholders. Clarion is an American company, based in Florida, which develops 4GL database application development tools for DOS, including Clarion Professional Developer and Clarion Report Writer. JPI is practically the last bastion of British-based mainstream compiler development. The company develops the TopSpeed family of languages comprising Pascal, Modula-2, C and C++ optimising compilers that share a common code generator and IDE.

JPI has been collaborating with Clarion for the last two years to include its compiler/link technology into V3.0 of Clarion's Professional Developer 4GL (due to ship in Q1 '92). The takeover comes as no surprise to industry-watchers who have recently speculated that JPI was spreading itself too thinly with so many compilers to support and maintain (at one time the company was planning to develop an Ada compiler as well).

JPI was founded in 1987 by Niels Jensen, a co-founder of Borland International, and five other former Borland developers. The development staff of JPI will remain in London and will be known as the TopSpeed Institute, which will continue to develop and maintain the company's language products. Financial details of the merger were not disclosed.

Still de-fluffing C

When Gimpel Software releases a new version of PC-lint - the program which takes all the fun out of C programming, by reporting the very code inconsistencies which make that language such a joy to use - we usually get to print a few lines of superficially ok (but actually hopelessly bug-ridden) code, so that You the Reader can test your expertise. The example provided with PC-lint 5.0, however, is too long for this. Never mind.

The two new features which Gimpel is emphasising are stronger type checking and an extreme sensitivity to uninitialised variables - the latter achieved by an algorithm which follows control flow through the code. Other listed improvements include a rude message when `floats` are tested for equality (you laugh, but I've known many a proper, *paid* programmer do this) and a 'handsome new manual sporting real section numbers'.

Gimpel is based in the US, so for UK prices we phoned Grey Matter (0364 53499) who quoted us £80 ex VAT for the DOS-OS/2 version and £130 for the (faster, larger capacity) 386 DOS-extender version.

The Muse of objects

Persistent Objects & Extended database Technology (POET) is a low-cost C++ object-oriented DBMS for DOS, Windows, Interactive UNIX and the NeXT operating environments. There is a single API for all platforms and database files are binary-compatible across the different systems.

Developed by Berlin-based software house BKS, the POET system implements C++ language extensions (such as `persistent`, `transient` and `index` keywords) through a pre-compiler which generates C++ v2.0 source code and header files. These are then compiled along with the user's application code by a native C++ compiler. The pre-compiler also supports templates for containers. In addition, there are various utilities supplied including a class browser (graphical under Windows) and a version control system for the class dictionary.

Complete database facilities are provided including complex selection through predefined query classes, sorting and navigation. The company is presently working on a multi-user client/server version of the product.

Currently, most object-oriented databases are available only on high-end UNIX systems and are often prohibitively expensive. The single-user POET development system for Windows costs just £299 (versions are available for Borland C++ 2.0 and Zortech C++ 3.0 compilers). It is distributed in the UK by Silicon River, a company recently set up by ex-Zortech directors and headed by former Zortech MD Paul Leathers. Silicon River is on 081 3177777.

One all

In a US court, Intel has failed to get an injunction against Cyrix, a manufacturer of competitively-priced 80x87 clone coprocessor chips, to prevent Cyrix from shipping its coprocessors. Part Two: Intel has succeeded in getting an injunction against ULSI, also a manufacturer of competitively-priced 80x87 clone coprocessor chips, preventing ULSI from shipping *its* coprocessors. Both disputes concern alleged infringements of Intel's patents.

System Architect not only has the power and flexibility to handle your most complex application, it is *easy to use* and *affordable*. As the only CASE tool to be specifically developed for MS Windows, System Architect brings you a graphics oriented, fully integrated multi-user development platform for PCs.

In addition to standard CASE tool features, System Architect offers a wealth of additional functionality.

Methodologies Supported

- SSADM
- Yourdon/DeMarco
- Ward & Mellor
- Object Oriented
- Gane & Sarson
- Information Engineering

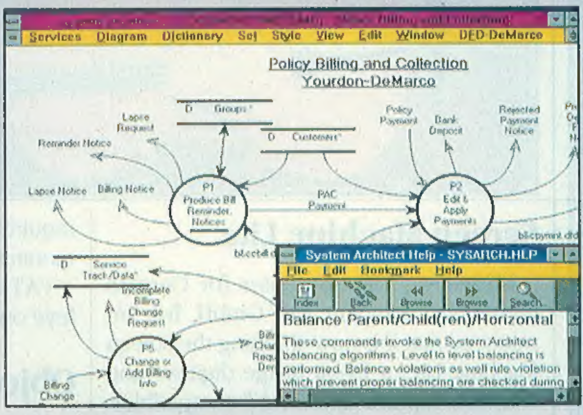
Full Data Modelling

Lifecycle Support

- Requirements Tracking
- Analysis/Design
- Prototyping
- Schema Generation
- Code Frame Generation

Benefits

- Ease of Use
- Readily Affordable
- Networking
- PC Based
- Increased Productivity



Balance Parent/Child(ren)/Horizontal

These commands invoke the System Architect balancing algorithms. Level to level balancing is performed. Balance violations as well rule violation which prevent proper balancing are checked during

To arrange a demonstration,
attend one of our regular System Architect seminars,
or for further information please phone us on

Tel: (0926) 450858

Fax: (0926) 422165

Supporting IBM's AD/Cycle

Real-Techniques and Methods Limited
118-120 Warwick Street, Royal Leamington Spa,
Warwickshire, England CV32 4QY.

Special Features

- User Definable Attributes
- Customisable Reporting System (Matrix and Tabular)
- Automated Documentation
- Import/Export/Extract/Merge
- Multiple Document Interface
- Import of External Graphics
- CRUD Matrices
- Mini-Spec Balancing
- OS/2 Support

Proven Reliability

- Over 6,500 licences installed. Users include:
  - NCR • Toshiba • Racal
  - IBM • Bull • Prudential

Training and Support

Full training and support for both the novice and advanced user. Also training courses in Structured Methods and Object Oriented approach.

MICROSOFT WINDOWS
Version 3.0 Compatible Product

System Architect™

CIRCLE NO. 428

Oil and Water

If you're having peculiar problems with code compiled with Borland C++ V3.0 and linked to a library generated by Borland C++ V2.0, then stop trying. Binaries produced by V3.0 are (potentially) incompatible with V2.0; you must recompile library source from scratch, or contact your library supplier for an upgrade. This applies to Borland's Turbo Vision and OWL libraries, too.

Real Time Lib++

Capsule is a new general purpose C++ class library from Microtec Research. It contains a selection of container classes, such as singly and doubly linked lists, circular lists, stacks, queues and heterogeneous linked lists etc. Capsule also provides error detection and application specific classes derived from these base classes, including classes to handle PostScript or graphics output and string/numeric conversions etc. Applications developed under Capsule can be debugged using Microtec's own Xray debugger. Capsule costs £390. Microtec is on 0256 57551.

Energize C/C++

The Energize system from Lucid Inc is a new Client/Server C/C++ programming tool set for SPARC workstations which integrates into the UNIX environment and provides object-oriented development tools. The tool set contains the GNU Emacs editor and the GNU debugger, as well as Lucid's own C and C++ compilers which provide incremental compiling. The Server part of Energize provides an object-oriented database for storing the all the program information when more than one developer is working on the code. Several browsing tools have also been included to simplify the navigation through code. Energize costs $3250 for a single user licence and is distributed by Lucid Inc on 010415 3298400.

Foxy CodeBase

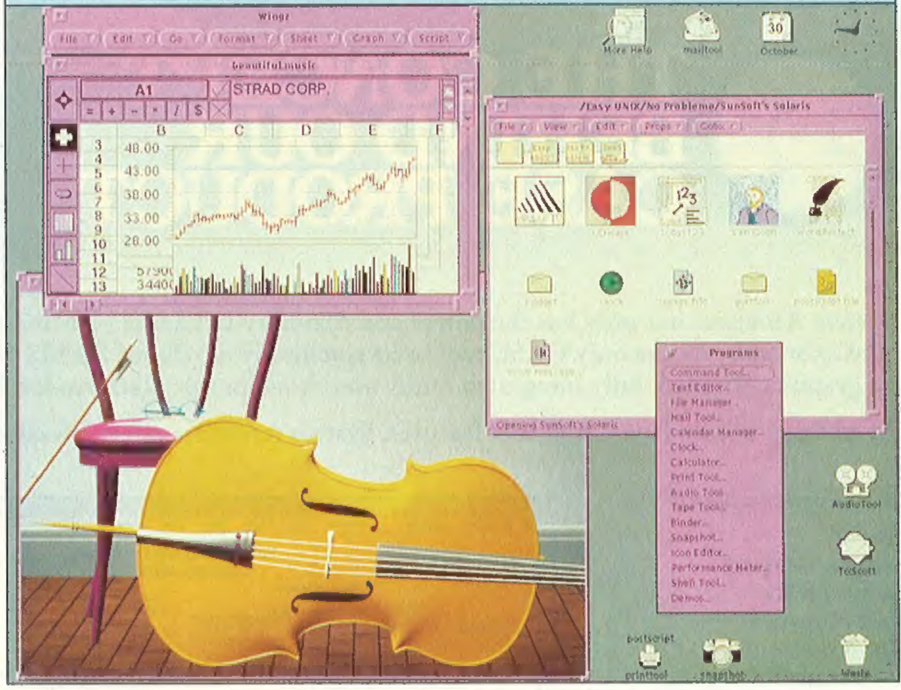
CodeBase V4.5 is the latest incarnation of Sequiter's multi-user database management library for C, C++, Visual Basic and Turbo Pascal for Windows. In addition to supporting dBASE, Clipper and FoxPro, CodeBase now supports dBASE IV and FoxPro V2.0 CDX compound index file formats. Fox's new CDX format significantly reduces the size of index files which means that less data has to be read or written to disk. This results in applications that run faster. CodeBase V4.5 costs £255 and is distributed in the UK by The Software Construction Company on 0763 244114.

Solaris

Sunsoft, the system software subsidiary of Sun Microsystems, has introduced a new 32-bit operating system called Solaris which will be available on both the Intel 80x86 and SPARC platforms. Applications developed for Solaris on these platforms will be source code compatible, enabling developers to write applications capable of running on both platforms (the source code will only need to be re-complied).

Solaris is based on SunOS V4.1.1 (which incorporates UNIX System V Release 4) and offers a 32-bit multi-tasking, multi-processing and multi-threading operating system. It provides an Open Look desktop with 14 applications including a file manager, networked calender and a mail application. The Solaris development environment includes a C compiler, XGL graphics libraries and 57 F3 fonts. Network administration is simplified using the ONC (Open Network Computing) protocols which can run on all major operating systems including DOS and VMS. This enables distributed applications to be developed across heterogeneous networks.

Solaris V1.0 for the SPARC platform is currently only available to OEMs. Sunsoft expects to ship Solaris V2.0 in the third quarter of 1992. This will support both Intel and SPARC platforms. For more information contact Sunsoft on 0494 472900.



ADVANCED BUSINESS GRAPHICS

Screen Machine Lite

Magnifeye, UK distributor for German multimedia company Fast GmbH, has announced that it is now shipping the Screen Machine Lite, a 24-bit image digitiser and real time capture card for AT-compatibles. The Lite comes with a complete image capture application called Screen Machine Photo which enables users to grab frames of video from almost any video source. The same software, which runs under Windows 3.0, allows the captured images to be cropped and compressed using the JPEG standard and converted into one of the 15 image formats supported (including TIFF, GIF, PCX, RLE and TGA). Colour balance and saturation can also be adjusted.

Screen Machine Lite supports NTSC, PAL, VHS and S-VHS so that camcorders, VCRs and laserdisks can all be used, including the Canon Ion still video camera. The card fits into a 16-bit slot in any AT-compatible and requires 2 MB RAM, a mouse and a VGA monitor. The card plus software costs £495 + VAT and can be purchased from Magnifeye on 071 2218024.

ObjectStore for Windows

Previously available only on UNIX platforms, ObjectStore object-oriented database management system (ODBMS) is now available in a Windows version. 1992 looks like being the year of the ODBMS with ObjectStore for Windows being the second Windows-hosted product to be released in the last two months (see POET elsewhere in News).

Developed by American company ObjectDesign, ObjectStore for Windows is available in standalone and LAN-based client/server versions. The product has three principal components: the run-time ODBMS, the application interface (API) and a set of C++ development tools.

Deep C Diving.
Without the lead boots.

When you've been in the Compiler business as long as we have you find out a few things. Like the importance of DOS and Windows Development Environments and DPMI support. That's why with half a million programmers already using Borland C and C++ products, our 3rd generation C/C++ compiler is about to make an even bigger splash than ever.

Borland C++ 3.0
£299.95 + VAT

Borland C++ 3.0
and Application
Frameworks
£439.95 + VAT

FREE copy of Norton Desktop for Windows.*

As well as all the features you would expect from a professional compiler, such as customisable Global Optimisation for both size and speed, AT&T 2.1 C++ and ANSI standard C compilers, you also get our prize winning Turbo Debugger for DOS and Windows, Turbo Profiler and Turbo Assembler.

And just so you won't get cold feet about diving into the world of C++, we've designed Application Frameworks, full of user interface objects that you can bolt right onto your code to create great looking applications under DOS or Windows. We've even included the source code.

And because we know the importance of fast development, the visual ObjectBrowser lets you see the structure of your code at a glance and navigate through classes, functions and variables to make changes without ever leaving the integrated Windows environment. The EasyWin library even lets you re-compile DOS applications to run under Windows without changing a line of code.

Turbo
C++ for
Windows
£119.95
+ VAT

Introducing Turbo C++ for Windows, the quickest way to create Windows Applications. With inbuilt tutorials, context sensitive help and ready to use examples, together with the ObjectWindows Library full of bolt on user interface objects now you can create Windows applications fast. For more information or for upgrade details, take a deep breath and **call FREE on 0800 212727** or complete and return the FREEPOST coupon today.

All Borland products are trademarks or registered trademarks of Borland International. Other brand product names are trademarks of their respective holders. Copyright © 1991 Borland International Inc.

*Free with Borland C++ 3.0 & Application Frameworks whilst stocks last.

BORLAND
SOFTWARE
KNOW HOW

Borland International (UK) Ltd
8 Pavilions, Ruscombe Business Park
Twyford, Berkshire RG10 9NN. Telephone 0800 212727

Name ____________________

Position ____________________

Company ____________________

Address ____________________

____

Postcode __________ Tel. __________

Please return this coupon to:
Borland International (UK) Ltd.
FREEPOST RG1571, TWYFORD, Berkshire RG10 8BR.
Telephone 0800 212727

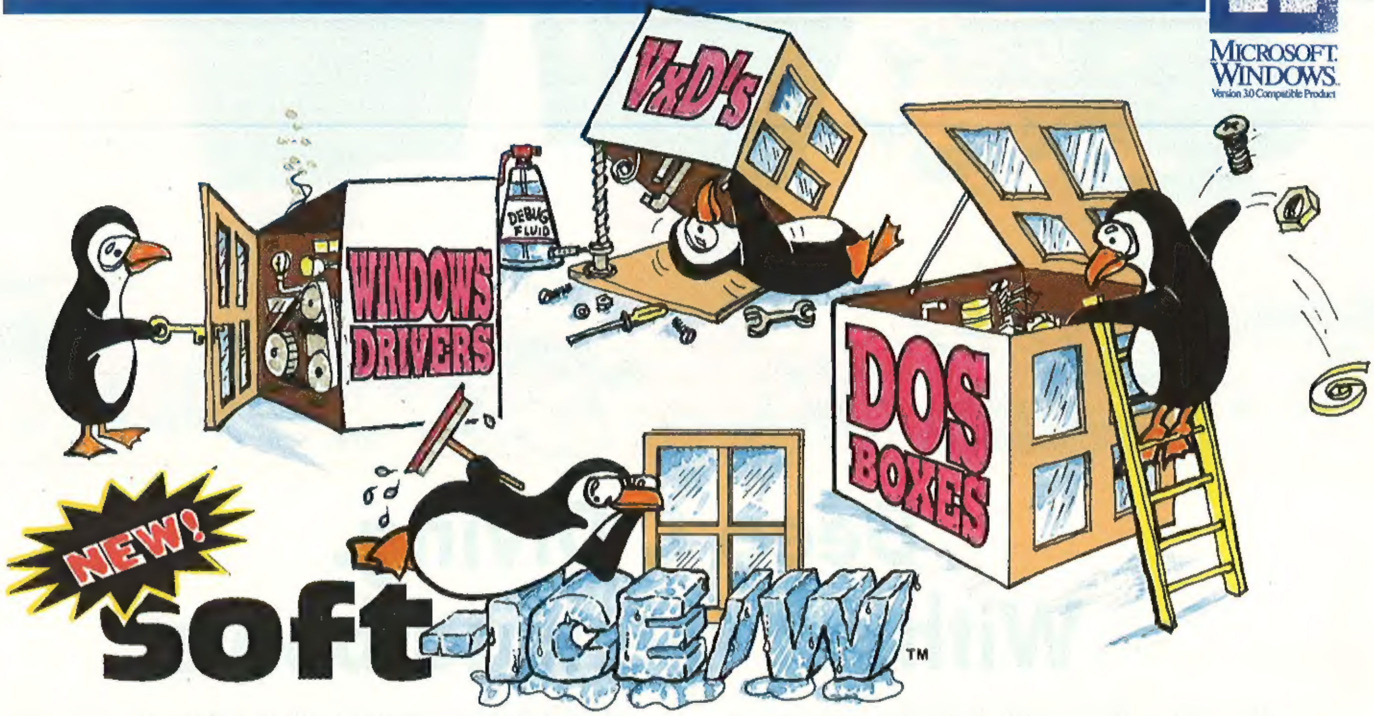
.EXE 2/92

CIRCLE NO. 429

™ Nu-Mega
TECHNOLOGIES INC

Get Inside WINDOWS!

MICROSOFT.
WINDOWS.
Version 3.0 Compatible Product



Debug Windows at the systems level!

Soft-ICE/W takes you inside Windows! Debug and explore with power and flexibility not found in any other Windows debugger! Soft-ICE/W allows you to debug at the systems or applications level or simply learn the inner workings of Windows.

- Debug VxD's, drivers and interrupt routines at source level
- Debug interactions between DOS T&SR's and Windows Apps
- Debug programs in DOS-boxes
- Display valuable system information (from the total memory occupied by a Windows application, to the complex internal structures of Windows)

Soft-ICE/W uses the 386/486 architecture to provide break point capabilities that normally require external hardware. Nu-Mega, which pioneered this technology with the introduction of its award winning Soft/ICE for DOS, now gives Windows programmers the same debugging power... *and still at a software price.*

Own the debugger that combines the best "view" of Windows internals with the most powerful break points of any software debugger.

Soft-ICE/W . . . Only £259.00 +VAT

CodeView for Windows users: see what you're debugging without flash. **CV/1 version 2.0** runs CodeView in a graphics window while viewing your application screen. Runs on <u>any</u> display that supports Windows.

CV/1 . . . Only £95.00 +VAT

WHAT THE EXPERTS ARE SAYING

"Soft-ICE for Windows is great! It helped me find, in fifteen minutes, a killer bug in a Windows virtual device driver that had eluded two people for several months. I can't see doing Windows development of any kind -- whether writing Windows applications, device drivers, or even DOS programs that have to run under Windows -- without it. In addition to being great for finding bugs, Soft-ICE for Windows has been essential for my work on a forthcoming book: on Undocumented Windows. Soft-ICE for Windows goes anywhere and does everything, so it's essential for anyone who wants to poke around inside Windows Enhanced mode. DOS programmers will find it a perfect way to learn how the Windows DOS extender and DPMI server work, and how Windows interacts with DOS. Windows Enhanced mode is the hacker's paradise of the 90s, and Soft-ICE for Windows is the tool that every serious Windows or DOS hacker will need. Nu-Mega has done a brilliant job!"

Andrew Schulman
Software Engineer, Phar Lap Software
Editor, Undocumented DOS
Coauthor, Undocumented Windows (forthcoming)

Tel: (071) 833 1022

Fax: (071) 837 6411

3-5 Cynthia Street,
London N1 9JF

Now in the UK
System Science

RISK = NULL
30 DAY
MONEY-BACK GUARANTEE

MICROSOFT WINDOWS IS A REGISTERED TRADEMARK OF MICROSOFT CORP. Soft-ICE/W AND CV/1 ARE TRADEMARKS OF NU-MEGA TECHNOLOGIES, INC.

CIRCLE NO. 430

The run-time environment performs all necessary object management and storage. The client/server implementation supports all NetBIOS-based PC networks (eg Novell, Banyan VINES and LAN Manager) and PC NFS for interoperability among PC and UNIX-based systems.

The API provides access to the run-time system using either a C or C++ library interface. All popular C and C++ compilers are supported including Borland C++, Microsoft C and Zortech C++.

Windows-hosted development tools consist of SchemaDesigner - a graphical database design tool, and The Browser - a tool for inspecting the contents of an ObjectStore database.

Unlike the POET ODBMS, ObjectStore for Windows doesn't come cheap. A standalone (single-user) development kit will leave a hole in your pocket the size of £3,495. Multi-user network versions start at £10,000. ObjectStore for Windows is available in the UK from Sema Software Technology on 0249 656194.

Faster Windows

The GainTEC GXA card is a new high-performance graphics adapter which is specifically designed to improve the performance of applications running in a GUI environment. The company claims that the card will deliver a 40% improvement over IBM's XGA card on Windows-based software, and is up to 10 times faster than most Super VGA cards. It plugs into either a 16-bit AT-type (ISA) slot or a 32-bit EISA slot and comes with 1 MB on-board VRAM.

Like IBM's XGA, the GXA provides dedicated hardware support for the major GUI functions using high-level commands, and thus eliminates most CPU and I/O intensive operations. All standard display resolutions are supported up to 1280 x 1024 in 16 colour mode, 1024 x 768 (non-interlaced) in 256 colours and 640.x 480 in 65,536 colours. A comprehensive range of software drivers is supplied with the card, providing support for most major GUIs, CAD systems and DOS applications. It is 100% VGA-compatible and offers full backward compatibility with CGA, HGC and MDA.

The GXA costs £345 + VAT and is available from Scene Double on 081 9583639.

DOMS

HyperDesk has unveiled its Distributed Object Management System (DOMS) which provides software developers with a set of tools for building and integrating distributed applications over a number of platforms including PCs, workstations and servers. Based on the Object Management Group's Object Request Broker (ORB) standard, DOMS is claimed to simplify object access in APIs, Remote Procedure Calls, network protocols and operating systems. It runs under both DOS and UNIX and supports Windows and Motif.

The HyperDesk Distributed Object Management System Development Kit costs $1995 (single user licence). Contact HyperDesk Corporation on 010508 3665050 for more information.

Dolphin Libs

The Dolphin C Toolkit is a C library for Microsoft C and Borland C/C++ which contains 225 miscellaneous functions including printer functions that interface with the DOS print spooler to allow background printing and functions for determining the type of disk present and for performing a software reboot. A Far Memory Manager and Debugging Library contains functions for analysing the far heap plus routines which dynamically allocate multi-dimensional arrays at run-time. The Dolphin C Toolkit costs $149 and the Far Memory Manager costs us $99. For more information contact Dolphin Software on 010510 4643009.

.EXE in 1992

You may notice some changes in this month's .EXE Magazine. Our readership survey suggested that C programmers wanted more C stuff, whereas non-C programmers wanted *anything* else. To try and accommodate everybody, we have replaced 'Code Page' with four new regular columns. '!C' (ie 'logical NOT C') is for practical, codey articles in anything except C; 'xBASE' is a *pied-à-terre* for the Foxy and the Clipperish and friends; 'Laine Stump's C++ Diary' will also be down-to-earth; and 'CUG' will provide a point of contact with the C User's Group, with the European C++ User's Group to follow soon. 'The Third Side' retires from our regular line up, although we hope to bring it back from time to time as a 'Special'. Stalwart columnists Peter Collinson and Jules May plough their lonely furrows as before.

When you have had a look at our efforts and cast this magazine aside in disgust, please write and tell us exactly why we should have left well alone, and anyway .EXE is not half as good as it was two years ago. Our *Letter of the Month* competition, now sponsored by Just Computer Books, offers a £20 book voucher as the prize for the best letter; writing fawning letters will not help you win it. Well, maybe.

Soft Mouse

NoMouse is a software replacement for the mouse, providing precise pixel control in Windows applications. It enables users of laptops and notebooks - not to mention those whose rodent has bust - to run Windows applications efficiently, without the need for a physical mouse. It includes the EasyFind utility which enlarges the standard mouse pointer, making it easy to see on LCD screens. It allows you to use short-cut keys to move quickly to the sides and corners of a window. NoMouse costs £39.95 and is distributed by GuildSoft on 0752 606200.

Affordable CASE

Personal-SELECT SSADM is a low-cost DOS-based CASE tool that provides support for both versions 3 and 4 of the SSADM methodology. The usual CASE features are all supported including data flow diagrams, entity life histories, logical data structures and, of course, a data dictionary. Versions are also available which support Yourdon and HOOD methodologies. Personal-SELECT costs £495 for a single-user version and £995 for a multi-user network version. It is published by SELECT Software on 0242 226553.

Watcom GO Pen

Watcom has announced an agreement with the GO Corporation, under which Watcom will supply the 32-bit C compiler for GO's PenPoint Software Development Kit. PenPoint is an object-oriented operating system, specifically designed for pen-based computing. The Watcom compiler is hosted under DOS or OS/2 and enables the development of applications for PenPoint on 32-bit Intel 80x86 systems. Version 1.0 of the PenPoint SDK is due to ship in the first quarter of this year.

Maastricht Conference

No, not another EC summit. The Fourth International Conference on Image Processing and its Applications (IPA), organised by the Institution of Electrical Engineers, will be held in Maastricht in the Netherlands from 7-9 April 1992.

More Shareware

We looked at EMS's C++ shareware library in our November '91 issue; now the company has introduced two new libraries: WINLIB (MS-Windows utilities, drivers, interesting bitmaps etc) and dBUtility (includes programs for dBASE III/IV, FoxPro, Clipper etc). Both libraries are US $149 ex delivery, call 0101 301 924 3594.

Letters

We welcome short letters on any subject that is relevant to software development. Please write to The Editor, .EXE Magazine, 10 Barley Mow Passage, Chiswick, London W4 4PH. Unless your letter is marked 'Not for Publication', it will be considered for inclusion in this section.

A Typical Reader...

Sir,

I was very pleased to read the results of the .EXE Readership Survey. Although I did not take part in it, as a relatively new subscriber, I should like to congratulate the team for the innovative way (disk based) and the effort gone into obtaining unbiased results.

Needless to say that I match the results of the survey to an uncomfortably close degree in practically every aspect, except that I am now 50 and currently unemployed.

I entered software engineering about 12 years ago from electronics by designing embedded system controllers and data acquisition systems, a skill for which demand fell off very rapidly.

Networking is now the name of the game, but without formal qualifications in Computer Studies I would have remained a 'grease-monkey' had it not been for my courage to enroll at Middlesex Poly for a fresh HND.

However, my recent efforts on job hunting have revealed that prospective employers prefer youngsters with minimal experience, although us old-timers, for not a lot more, would seem to offer much better value for money.

Since the bulk of your readership is middle-aged, it would be a very interesting to research into the future prospects of 45+ ages by finding out how the existing ones are doing.

Last, but not least, I should like to congratulate all the staff for the excellent presentation of the magazine and especially for the editorial manner adopted, which makes everything fun to read.

The introduction of classified ads will benefit both the magazine and the industry it serves, especially during these hard days.

Mr C Papazissimou
London NW2

OS/2 Babel

Sir,

I read with interest BJ Thomson's article on character sets *(Software Sans Frontières, .EXE December 1991)*. I have two points to make.

First, OS/2 PM programmers do **not** need to write their own equivalents for the `AnsiNext()` and `AnsiPrev()` functions - OS/2 comes with two functions, `WinNextChar()` and `WinPrevChar()`, to do this. OS/2 also has `WinUpperChar()` and `WinUpper()` which convert a single character or a string of characters into upper case. For some unknown reason, however, there is no lower case equivalent!

Second, one problem which was not explicitly mentioned was the input and display of numbers with decimal points. We English speakers are used to using a period ('.'), but many other languages use a comma. Writers of programs to be portable to other countries need to beware of code such as:

```
sscanf( ipbuf, "%f",
                    &fval);
```

since `sscanf()` will expect the input buffer to contain a period and will not process input containing a comma correctly.

As a further complication, other languages use the period, or some other character, where we use a comma to separate thousands. So, for example, 1024 can be written as '1,024' in English, whereas natives of many other countries in Europe would prefer '1.024'.

It is a shame that, as far as I am aware, no standard PM function exists to provide the equivalent of `WinQueryDlgItemShort()` for floating point numbers. Perhaps systems developers stick to integers!

Roger Orr
London

Stepstone

Sir,

Your recent article 'Of Bullets and Muskets' *(.EXE, October 1991)* by Al Roth, printed information that was not only erroneous, but extremely damaging to the reputation of The Stepstone Corp.

In this article Dr Brad Cox, ex-Chief Technical Officer of Stepstone, was quoted as saying 'Stepstone is essentially out of business... it's reasonably public.' At no time was Stepstone out of business.

In April 1991, Stepstone went through a major financial reorganisation. Since that time, for the first time in the history of the company, Stepstone is surviving without the funding of venture capital groups. We continued to support our customer base throughout the reorganisation.

We have hired back ex-employees, and are announcing three new products in the first quarter of 1992; Objective-C for the Macintosh, the Objective-C Interpreter for the SUNs, and ICpak 101 for the NeXT station.

The enthusiasm of our employees and their belief in the superior technical attributes of our products has kept our company alive and growing.

K K Tan
President
The Stepstone Corporation
USA

We apologise for printing potentially misleading information, and are happy to set the record straight - Ed.

QEMM saga #5

Sir,

I am able to confirm the claim by the Quarterdeck MD *(.EXE Letters, Nov '91)* that users keep stumbling over 'unpromoted' features. However, these may not be quite as fabulous as he believes.

A year ago I returned a Quarterdeck 'more info' coupon and ringed several Bingo cards, all to no avail. It took a mildly sarcastic letter to obtain information on QEMM and DESQview. The products were ordered, but on delivery a disk proved faulty. So did a replacement set. The third set was OK but the auto-install was not. It made various errors, some as elementary as copying QEMM386.SYS to a \QEMM directory, but omitting this path from its CONFIG.SYS modifications. Other errors were less obvious. The machine was a new, 'clean' 386 with 8 MB RAM so there were no complications. Now I have found that Quarterdeck has no record of my registration card. A letter to Quarterdeck setting out the above in more detail has yet to bring a response.

What of the products themselves? No complaints about QEMM, once the install errors had been traced and corrected. The clarity of the manuals is not helped by the excessive repetition of the company brand names. Over 20 per page is typical, on one page I have noticed 26 mentions. DESQview has proved less useful than anticipated. Perhaps I could use it more often - I should like to use it with Windows without it crashing on exit - but the time spent removing the wrinkles often becomes greater than the maximum time it could possibly save in actual use. Investments, whether of time or money, need to be justified by results.

Mr Poel says he would like .EXE readers to be more positive. Perhaps he could set us an example first?

Bob Rimmington
Burgess Hill

DR-DOS 6.0 is ok

Sir,

I am writing to you with reference to Ian Butterworth's letter *(.EXE Nov '91).* His client had apparently experienced problems running Clipper Summer '87 programs under DR-DOS 6.0. I find this rather puzzling. My company, A C K Data, has several applications amounting to around 5 MB of source code, all in Summer '87, which have been running under DR-DOS V6.0 for the last month with no problems whatsoever.

I wonder if Mr Butterworth's clients' computer was running with data compression? Our systems are set up with no data compression, but we are running the disk cache and task-switcher. In a computer with more than 1 MB RAM, DR-DOS will use the expanded memory, or will set up extended memory to look like expanded memory (using EMM386.SYS or EMMXMA.SYS), depending on the options selected during installation. If expanded memory exists, Clipper software will use it - potentially causing a conflict with software such as the disk cache or task-switcher. Stop Clipper using expanded memory by entering

```
SET CLIPPER=E0
```

at the DOS prompt. Normally, use of LIM memory should not cause this sort of problem - it's just that Clipper Summer '87 (I suspect) may not be too well behaved in this area.

I would be interested to hear from anyone else who may have experienced difficulties running either Summer '87 or V5.01 under DR-DOS V6.0.

Michael Towle
A C K Data
Nottingham

We also understand that Summer '87's handling of expanded memory is bugged - Ed

Bugwatch

Sir,

Further to Dave Riley's letter *(.EXE, June 1991)*, I have tracked down another potentially dangerous bug in the Microsoft C 'optimising' compiler V6.0. The bug only occurs with optimisation on (default) and is demonstrated by the following program (small memory model):

```
#include <stdio.h>

static char output, input = 4;

void main(void)

{
  unsigned x;
  int  fd, fh, fm, fs;
  char da, ho, mi, se;

  da = (char) (x + 1);
  ho = (char) (x + 2);
  mi = (char) (x + 3);
  se = (char) (x + 4);

  output =
        (char) ( 10 * input );
  printf("%d", (int)output);

  fs = fd + se;
  fm = mi + fs
  fh = ho + fm;
}
```

The program should of course print out 40, but the value that actually gets output is unpredictable and depends on the contents of the DH register when `main()` is entered.

Examination of the assembly code produced by the compiler reveals that the optimiser correctly 'ignores' the first four and last three instructions (since they only operate on local data they are redundant). If these lines are taken out of the source file, however, the compiler generates correct code! The error occurs because the compiler is trying to use a 'clever' method of multiplying by 10, thus avoiding use of the `MUL` instruction. Although using the `MUL` instruction would result in faster and smaller code for a 286 or 386 processor, the compiler still insists on doing it this way when the /G2 switch (generate 286 code) is used!

The moral of this and the conclusion we have come to here is always to disable optimisation when using MS C V6.0.

Patrick Johns
AGB International
London

PCL #1

Sir,

My December copy of .EXE was accompanied by a subscription invitation. Hmm! Possible, until I read the rubbish offered by Peter Flynn as a review of PCL.

We have software running in over 20 countries using the PCL run-time environment, no licence issues here. All of it uses communications, via PSTN, Public Packet Switching, in-house PADS, and the like... Peter Flynn 'wasn't able to make it (the terminal emulator) work anywhere.'

His is the sort of review that gives 'academic computing' a bad name in the commercial environment.

My subscription? Take it out of the fee that you paid for three pages of shredder fodder.

Russell Hunt
Managing Director
Applications Systems
Hampshire

PCL #2

Sir,

I read with much interest your review of the PCL compiler/interpreter in the December issue. I've been using PCL regularly since 1988, both for my work with IPS and for the independent work I do for some other news organisations. I've all but thrown out my other compilers because PCL is so fast and easy to use and has assembler functions to do the nasty stuff.

The main comment I would like to make to the review's author is that the Terminal function in PCL has **never** failed me. Most of my work involves RS-232 communications, so almost the first thing I do when I sit down in front of a client's computer is run Terminal and look at the serial ports. It **always** works, just like everything else with PCL, and it's dead easy. I can't imagine what he was doing wrong unless he had nothing connected to the port.

I do, however, heartily agree with him that it is a shame PCL doesn't allow the generation of .EXE and .COM files. But for his information, and that of your readers, I should point out that the run-time module of PCL may be distributed without any license fees, as the PCL documentation points out.

I have done this numerous times and once - when I ran up against an early Tulip 286 on which it wouldn't run (and which Tulip later admitted was making some spurious BIOS calls) - I had recourse to the Calend help line and found it offered straight away the information I needed. They only found it natural that I had distributed the run-time module to my client.

John Clements
Technical Director
IPS
London

We apologise for the inaccuracies in the PCL article - Ed.

Hypermedia

Sir,

I was interested to read the article by Dr Pickard, and also the letter published in your column by Jules May *(.EXE Nov '91 and Dec '91).*

It is our belief that hypermedia, and applications of these techniques to communications, will have a tremendous impact on our everyday lives and I found both Richard Pickard and Jules May's articles very thought provoking.

However, it is not easy to distinguish between teaching and learning or entertainment if you take arguments used when discussing hypermedia to their logical conclusion.

After all, we have only recently finished riding the wave of consumerism which has fuelled the development of the personal computer and funded practically every other activity in our modern day society.

In a consumer society, training is just a way of instructing its members how to produce more efficiently. Advertisements are just a way of training its members how to consume products more efficiently.

Most entertainment on our TV screens each night is merely the soft selling of products, even if the goods are not paraded before us as they are in game shows. The values we must aspire to as good consumers are certainly put before us or suggested in everything from *EastEnders* to *The South Bank Show.*

For the last 40 years we have gradually become what we consume, and the communications industry has grown fat by feeding off the consumers' and producers' need to communicate. It will, therefore, be interesting to see what effect multimedia and hypermedia has on our society.

Peter Kruger
Digithurst Ltd
Herts

Letter of the Month

We have a hot, new sponsorship deal for 1992! The writer of the best letter of the month, as judged by the Editor, will receive a £20 Book Voucher, courtesy of Just Computer Books. The best letter is the one printed first. Please note that letters submitted to this page may be edited.

Why Choose?

Between Relational and Network when Raima Data Manager has both and now MUCH MORE

The Raima Data Manager dbms is intended for use by the professional C applications developer. Raima is written in C and provides a complete set of sophisticated development tools that feature:

- High speed access to large or complex data
- Mainframe functionality
- Portability to any C environment
- Royalty-free run-time distribution
- Source code available in C.

Raima provides both Relational and Network model technology for programming flexibility. Retrieve a record using the relational keyed access method and all related records can be immediately available using the network database model.

db_QUERY provides SQL-like access to **Raima**, callable from within an application, **db_REVISE** allows restructuring of live databases.

Raima is fast. Access times for retrieving data are largely independent of the size of the database.

Raima is portable. Develop an application in any environment and your database access code can be ported without change to any other supported environment. Currently, **Raima** supports MSDOS, MS Windows, OS/2, QNX, UNIX System V and Berkeley, Ultrix and VMS.

Raima Data Manager dbms is a product of Raima Corporation and has been sold to thousands of C programmers in over 50 countries. **Raima** is fully supported in the UK and Eire by Systemstar Ltd. Comprehensive training is available together with bespoke system development. At Systemstar we offer **C-scape 3®** screen interface library with Look and Feel™ from Liant Software Corporation to provide **Raima** programmers with a complete application development environment.

For more information about Raima Data Manager dbms, db_QUERY, db_REVISE and C-scape 3 call Systemstar in Hertford on (0992) 500919.

FEATURES	YES	NO
Single and multi-user available	✓	
Relational B-tree indexing	✓	
Network database model	✓	
Multiple database access	✓	
Referential integrity	✓	
Transaction processing	✓	
Automatic recovery	✓	
Record and file locking	✓	
RAM resident		✓
Relational Query and report writer	✓	
Total database redesign/restructuring	✓	
C compilers: most supported	✓	
C ++ compatible	✓	
Operating Systems: VMS, Ultrix,	✓	
UNIX, BSD, SunOs, QNX	✓	
MSDOS, OS/2	✓	
Windows 3	✓	
LANs: Netbios and NFS	✓	
Read and write WKS, WK1 and DBF files*	✓	
Source Code available	✓	
Training courses available	✓	
Run-time Royalties (Absolutely NOT)		✓
*using WKS Library		

NEW SHOWCASE

Support for NFS
Object Manager Class Library
Protected Mode for **MS-DOS**
VISUAL BASIC support for Windows

SYSTEMSTAR
L I M I T E D

1-3 PARLIAMENT SQUARE HERTFORD SG14 1EX
TELEPHONE: (0992) 500919 FACSIMILE: (0992) 554261

CIRCLE NO. 432

Practically browsing with Clipper 5.01

Nantucket's Clipper 5.01 is worth getting to grips with if you're prepared to leave your old xBASE code behind. James Ormrod shows why with a configurable, saveable TBrowse.

Nantucket's Clipper 5.01 has proved to be a bit of an enigma: less than fully object oriented, more than just another xBASE product, a little C-like and not at all like dBASE, the Clipper '87 developer moving to 5.01 needs to do more than just assimilate extra features and functions - he needs to absorb a whole new philosophy and probably change their programming style.

The key to making the move to 5.01 is to 'think modular', to enlist 5.01's new features as allies in the battle against complexity, and to reject some of the old Summer '87 features.

Why bother? As always, it depends on your needs. If you want to knock something up for a one-off user quickly use a 4GL or grab a copy of Dataease or Foxbase. If you want to construct a core business or complex application that will have a wide distribution and a long life, use Clipper 5.01. The resulting system will be robust, fast, easily maintainable and extensible. If you should then go on to develop more applications you'll be able to reuse large chunks from the previous one, a situation that gets better and better the more applications you develop.

Reusability has been described as the 'Holy Grail' of software development, but is actually a by-product of modular design. If you 'think modular' your code will automatically be reusable. The rest of this article looks at combining Clipper 5.01's new features with modular design to build a reusable file browser with an object-oriented flavour.

5.01 Concepts

5.01 presents the new user with a bewildering array of new concepts, many of which are more familiar to a C programmer than to a Summer '87 user. It's easy to be put off and to rely on familiar techniques. Many developers have still not moved to 5.01 and of those who have, the majority are still writing Summer '87 code. This is a pity because 5.01 is not very good at compiling Summer '87 code and is in any case a far superior development language. The major concepts used in the code presented in this article are covered below. Figure 1 illustrates the pre-processor, including an example of a code block.

Pre-processor - A batch-driven 'search-and-replace' feature. Patterns to match and

```
//Include Nantucket definitions for fopen()/fcreate():

#include "fileio.ch"

/*
 #xcommand defines an extended user
 defined command which cannot
 be abbreviated to 4 characters:
   default cName to ""
 becomes:
  cName := iif(cName == NIL, "", cName)
*/
#xcommand DEFAULT <p> TO ;
          >                ;
         <p>:=iif(<p> == NIL,,<p>)

/*
 #xtranslate is similar to #xcommand
 but is commonly used where syntax will
 be "function-like". The code block
 illustrated has a parameter passed to
 it, cFile, which is assigned to the
 variable cF. cF is visible only within
 the code block. The block returns the
 result of evaluating the if() in this
 case a file handle.

 nHnd := OpenOrCreate("error.log")
Will produce:
 nHnd := eval({|cF|                ;
              if(file(cF),         ;
              fopen(cF, FO_READWRITE),;
               fcreate(cF, FC_NORMAL)} ;
              },"error.log")
*/

#xtranslate OpenOrCreate(<cFile>)    ;
      =>                             ;
      eval({|cF|                     ;
            if(file(cF),             ;
            fopen(cF, FO_READWRITE),;
            fcreate(cF, FC_NORMAL)} ;
           }, <cFile>)

/*
 #define is used for constants and
 volatile data. It is the only
 directive that IS case sensitive:
*/
#define BUFFSIZE 1024

// Sample code using the directives

function audittrail(cLog)

local nHnd,;
      cBuffer := space(BUFFSIZE)

default cLog to "audit.log"

nHnd := OpenOrCreate(cLog)

// Etc.

return NIL
```

Figure 1 - Example pre-processor directives including a code block.

replace are specified either at the top of the source file or in a separate #include file. Pattern matching is performed before the program is compiled. The pre-processor has three main uses; it can enhance program readability, it can separate volatile data from the rest of the system making changes easier, and it can hide complexity by mapping user defined commands onto function calls (all commands except flow-control commands in Clipper are mapped to function calls in STD.CH).

Code Blocks - A mechanism by which a variable can point to executable code. The code can be executed directly through one of 5.01's EVAL() functions or indirectly via an object's method. A Code Block is a little like a C pointer-to-function and can have parameters passed to it and return a value.

Multidimensional arrays - An array in 5.01 is more like an undeclared structure than the sort of arrays C programmers are used to: any element in any array can have any data type, including another array. Array names point to the array structure in a similar manner to the way pointers work in C, so the same array can be referenced by more than one variable, and references to arrays can be passed to and returned from functions without the array being copied.

TBrowse object - an instance of the TBrowse class that contains the data (instance variables) and code (methods) required to define and manipulate a browse table. A TBrowse object relies on one or more TBcolumn objects, each of which describes the formatting, display and contents of a single table column. There are two other fixed classes in 5.01 - Get objects are used to control data-input and an Error object is instantiated when a run-time error occurs.

Static and local scopes - A local variable is visible only within the function in which it is declared and is destroyed when the function returns. A static variable has local scope, but is retained after the function returns, coming back into scope when the function is called again. A file-wide static variable is declared outside a function and is visible to all functions within the same source (.PRG) file. A static function may only be called by functions declared within the same source file. These scopes facilitate encapsulation by allowing the developer to hide data more efficiently than the PUBLIC and PRIVATE scopes of Clipper '87.

The application

A common technique in database applications is to 'browse' a table of data. In its simplest form this involves displaying part or all of a DBF file, each field corresponding to a column in the table. The user can then cursor through the table to view the contents of the file. Clipper Summer '87 extended this concept with the DBEDIT() function, which allowed the developer to specify the contents and formatting of each column and passed control to a configurable key-handler whenever a key was pressed, making it possible to build an entire user-interface around a browse table.

```
#include "box.ch"
#include "inkey.ch"
function main()
local i       ,; // Array index
      oTB     ,; // Tbrowse object
      oTC     ,; // TB column object
      cFld    ,; // Field name
      bGSet   ,; // Field get/set block
      nKP := 0   // Code of key pressed
/*
 Open the file to browse &
 instantiate the TBrowse object:
*/
oTB := TBrowseDB(02, 10, 23, 70)
oTB:headSep:=chr(196)+chr(194)+chr(196)
oTB:colSep :=" " + chr(179) + " "
oTB:footSep:=chr(196)+chr(193)+chr(196)
use staff index empcode
/*
Instantiate a TBcolumn object for each
field & add it to the TBrowse object:
*/
for i := 1 to fcount()
/*
 fieldname(n) returns the name of field
 number n. fieldblock() returns a
 "get/set" code block for the field,
 which when EVALuated gets the value of
 the field & sets it to the new value
 if one has been passed.
*/
  cFld  := fieldname(i)
  bGSet := fieldblock(cFld)
  oTC   := TBcolumnNew(cFld, bGSet)
   oTB:addcolumn(oTC)
 next
 /*
 Set the screen up & then pass
 control to the key-handler:
 */
 cls
 @ 01, 09, 24, 71 box B_SINGLE
 do while (nKP := inkey()) != K_ESC
 /*
  oTB:stabilize works incrementally &
  can be interrupted by a keystroke, so
  you don't have to wait for the screen
  to redraw every time you press a key.
 */
  do while ((! oTB:stabilize())    ;
     .and. (nKp := inkey()) == 0)
  enddo
/*
  We used TBrowseDB() so these methods
  have been set up for us (this is just
  a sample):
 */
   do case
   case nKP == K_RIGHT;  oTB:right()
   case nKP == K_UP;     oTB:up()
   case nKP == K_LEFT;   oTB:left()
   case nKP == K_DOWN;   oTB:down()
   case nKP == K_HOME;   oTB:home()
   case nKP == K_PGDN;   oTB:pagedown()
   case nKP == K_END;    oTB:end()
   case nKP == K_PGUP;   oTB:pageup()
   endcase
 enddo
return NIL
```

Figure 2 - Basic TBrowse

5.01's browsing abilities are even more powerful, using an object-oriented approach to make every aspect of the browse configurable. A TBrowse object can be configured to browse anything, in any order, including arrays, DBF files and even binary files. Once configured, columns in the object can be frozen or hidden spreadsheet-style, re-sized, swapped or removed without exiting the browse. Each column can have its own colour scheme and individual cells or groups of cells can be coloured. Cells can even be coloured according to their content (negative numbers appear red, for example). The key-handler for a TBrowse object is also totally user definable (DBEDIT() used to intercept certain keys for its own use) and you can even set keys up to behave differently for each column in the table.

A TBrowse object is configured by changing its exported instances, which can also be interrogated. The ability to interrogate instances means the TBrowse object could be reconstructed from copies of its instances, which could be stored off-line and re-loaded when required. Following on from this, it should be possible to write a routine that allows a user to configure a TBrowse to his satisfaction and then to save it to a disk file, from which it could be reloaded. Such a routine could be included in shipped applications to allow power-users the flexibility to change their user interface or used as a productivity or prototyping tool by developers.

There is a catch (of course!) Neither Clipper objects nor code-blocks can be directly stored off-line, so some translation will be required.

Configuring a browse table

The steps involved in configuring and invoking a TBrowse are as follows:

Instantiate the TBrowse object with a call to TBrowseNew().

Initialise instance variables to control colour, heading and column separators etc.

Define and assign methods to control cursor movement etc.

Instantiate a column object for each column in the browse table.

Change the column's instance variables to control width etc, if the default is not acceptable.

Define and assign a method to control the data source for each column.

THE ULTIMATE DEVELOPMENT MACHINE.

If you could create your own computer for developing software what would you build?

You'd need a system to increase your productivity, one that helped you create applications in half the time it takes you now – and complete projects ahead of schedule!

You'd want a true multitasking system with virtual memory. Because that would give you the kind of computer that will let you compile a program in one window while debugging in another, so if your new application should crash your whole system won't die.

You'd give it plenty of power and speed to handle even the most complex projects, complete with UNIX and the Motorola 68040 processor.

With your users in mind, you'd want an Interface Builder so you could create an outstanding user interface – quickly and easily.

And you'd want an object oriented programming environment so that you could reuse your software components and it would help if it came with a complete object library.

Fortunately, there's a computer that already offers this, the NeXT computer. The tools provided by the NeXTstep operating environment make it easy to develop advanced, user-oriented programs.

So take the NeXTstep and call 081-565 0005 for more information, or fill out the coupon.

The Professional's Workstation

I'd like to receive more information on NeXT □

I'd like to receive more information on NeXT Developer Courses □

Please complete and return to: NeXT, 286 Bath Road, West Drayton, Middlesex UB7 0DQ.

Name:

Company:

Address:

County:

Telephone: Post code:

.EXE 2/92

CIRCLE NO. 433

Add the columns to the TBrowse object.

Stabilise the TBrowse object (which draws it on the screen).

Pass control to the key-handler (usually a CASE statement).

Most browse tables manipulate data from DBF files and TBrowseDB() is supplied to instantiate an object with default cursor movement methods for these.

Figure 2 illustrates the steps involved in setting up a basic browse for a DBF table.

Having configured the TBrowse object, our aim is to be able to save it to disk. Since we can't write the object to disk directly, we must find a way of saving its unique instances and methods so that we can reconstruct it by using them to configure a new object. Objects in 5.01 are physically implemented as structures which we can model using arrays.

Saving and restoring arrays

Clipper lacks functions to save and restore arrays to and from disk files, so we will need to write them. These will be useful additions to our armoury, not least because they can be used as alternatives to the 5.01 SAVE and RESTORE commands (which write/read variables to/from disk .MEM files), which only work with PUBLIC and PRIVATE variables.

Code for a save array function is illustrated in Figure 3. The disk file created has an offset table at the top with the data following it. Each array has an entry in the offset table with the array name and the offset into the file at which its data starts, measured from the start of its offset table entry.

As new arrays are added the file grows in both directions; the new offset table entry is inserted at the very top of the file and the data is appended to the bottom. This mechanism ensures that the offsets of existing arrays do not have to be adjusted when new arrays are added to the file.

Not all Clipper data types can be saved in this manner: references to code blocks are created at run-time and cannot be stored off-line. Fortunately 5.01 includes a means of compiling code blocks that have been created as strings ('stringified blocks') using the & symbol, which will allow us to save and retrieve a block as a string and compile it later.

Saving a TBrowse object

An object in Clipper 5.01 can be 'converted' to an array by adding an element to the array for every exported instance variable.

Methods are stored as stringified blocks and compiled to code blocks when the object is restored. Information external to the object that is required to recreate the browse table will also be stored in the array, including the names of the DBF and index files, the

```
#include "fileio.ch"
#include "inkey.ch"

function SaveArray(aArray, cName, cFile)

 local nHnd,; // File handle
       nOff,; // File offset of array
                cOff   // nOff as 4-byte int

// Open or create save file:
 nHnd := OpenOrCreate(cFile)
 if nHnd == F_ERROR
    qout("Can't open/create: " + cFile)
    return F_ERROR
 endif

// Calculate data offset:
 nOff := fseek(nHnd, 0, FS_END) + 16
 cOff := l2Bin(nOff)

// Add jump table entry:
 nHnd := AddOff(nHnd,cName,cOff,cFile)

// Convert array to string & append it:
 fwrite(nHnd, Arr2String(aArray))
 fclose(nHnd)

return 0

function AddOff(nHnd,cArr,cOff,cFile)

 local cBuf := space(512),;// Buffer
       nThn              ,;// Tmp file
       nByte               // Bytes read

// Create & open temporary file:
 nTnd := fcreate("TEMP", FC_NORMAL)

// Write offset table entry:
 fwrite(nTnd, padr(cArr, 12) + cOff

// Append the old file:
 fseek(nHnd, 0, FS_SET)
 do while (nByte := fread(nHnd, @cBuf,;
                          512) > 0)
    fwrite(nTHnd, cBuf, nByte)
 enddo

// Rename the new and re-open it:
 fclose(cTHnd)
 fclose(cHnd)
 ferase(cFile)
 frename("TEMP", cFile)
 nHnd := fopen(cFile, FO_READWRITE)
 fseek(hnd, 0, FS_END)
return nHnd
```

```
function Arr2String(aAr, cBuf)


 local nLen  ,; // Number of elements
       i     ,; // Array index
       cN    ,; // Number as a string
       cEtype,; // Element data type

/*
  cBuf is passed back by reference in
  recursions & needs to be initialised
  on the first call:
*/
 if cBuf == NIL
   cBuf := " "
 endif

/*
  Calculate number of elements in array
  & append type & length to buffer:
*/
 nLen := len(aAr)
 cBuf += valtype(aAr) + i2bin(nLen)


/*
  If the element is an array, recurse
  Otherwise, append its data type, then
  append its length & data if it is a
  simple data type (C, N, D or L).
*/

for i := 1 to nLen
 cEtype := valtype(aAr[i])
 if cEtype $ "AO"
   Arr2String(aAr[i], @cBuf)
 else
   cBuf += cEtype
   do case
   case cEtype == "N"
     tmp := ltrim(str(aAr[i]))
     cBuf += i2bin(len(tmp)) + tmp
   case cEtype == "C"
     cBuf += i2bin(len(aAr[i])) + aAr[i]
   case cEtype == "D"
     cBuf += dtoc(aAr[i])
   case cEtype == "L"
     cBuf += iif(aAr[i], "T", "L")
   endcase
 endif
next

return cBuf
```

Figure 3 - Saving a multidimensional array

● QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR ●

THE COMPLETE PICTURE FOR C & C++ TRAINING

A fully comprehensive range of hands-on workshops

FOR PROSPECTIVE C PROGRAMMERS ...

Introduction to C Programming & Design

This course teaches C to new programmers, focusing on good programming practice and techniques.

Duration: 5 days **Cost:** £900
Frequency: Monthly

FOR NOVICE C PROGRAMMERS ...

C Booster

This hands-on workshop teaches the tools and understanding required for a broader knowledge of C.

Duration: 2 days **Cost:** £400
Frequency: Monthly

FOR HIGH LEVEL LANGUAGE PROGRAMMERS ...

C Programming Workshop

The course provides a hands-on introduction to C, teaching the syntax and semantics of the language.

Duration: 5 days **Cost:** £950
Frequency: Weekly

The Instruction Set has the right course for you

FOR EXPERIENCED C PROGRAMMERS ...

Advanced Programming in C

This course addresses the more advanced aspects of C and advanced programming techniques, including implementation of abstract data structures.

Duration: 5 days **Cost:** £1000
Frequency: Monthly

FOR EXPERIENCED C PROGRAMMERS ...

C++ Programming Workshop

This introduction to C++ teaches the constructs of the language and emphasises the use of object oriented techniques.

Duration: 5 days **Cost:** £990
Frequency: Monthly

Save Up To 20%

Series Booking, Complementary Courses Booking, Early Booking and Partnership Programmes make it easier for you to take advantage of premium quality training at a reduced price.

THE INSTRUCTION SET
Hoskyns Open Systems Training
071 253 5121

□ I would like to reserve a place on:

Course Title: ____________________

Please send me a course outline, a training schedule and booking form

□ I would like more information on the above courses

□ I am also interested in other training fron The Insruction Set:

□ UNIX
□ AIX
□ Communications & Networking
□ Windows
□ Programming Languages
□ Object Oriented Technology
□ Relational Databases
□ OS/2

□ Please phone me to discuss my training requirements

Name ____________________
Position ____________________
Company ____________________
Address ____________________
____
____
Telephone ____________________

Clip the coupon and return to:
The Instruction Set, FREEPOST (NW3832), London, EC1B 1DP
Or fax us on: 071-251 2853
Any enquiries please call us on 071-253 5121

The Instruction Set acknowledges all registered trademarks.

.EXE 2/92

● QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR ●

QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR

QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR ● QUALITY TRAINING EVERY WEEK OF THE YEAR

CIRCLE NO. 444

type of box that surrounds the table and the key-handler.

In the key-handler in Figure 2, a `CASE` statement is used to determine the code run when a key is pressed. In Clipper, each key has an 'inkey' value - the integer returned by the `inkey()` function when that key is pressed. Inkey values can be positive or negative; for example the inkey value of the F10 key is -9, the value of the RETURN key is 13. We could write a more flexible version of the key handler by creating a two-dimensional array of code blocks, the first dimension representing positive inkey values and the second negative values.

When a key is pressed we look it up in the array and evaluate the appropriate code block. If we need to change the action of an existing key, or define a new key to perform a new action we can construct a code block and assign it to the appropriate element in the array - the key handler will do the rest.

To go one step further, we could store stringified blocks in the array instead of blocks; that way we could store the key handler methods off-line with the object.

In recognition of the need for an object to be able to carry around more than the pre-defined instances and methods, Nantucket has provided each class with a cargo slot that can have any data type written to it. The cargo may hold simple data, a code block, another object or even a multi-dimensional array.

Figure 4 shows code that illustrates the new key handler and the use of the cargo slot to store miscellaneous data. The key handler methods array contains stringified code blocks which are 'compiled' within the key handler. In a 'production' version the contents of the array would be pre-compiled outside the key handler for speed.

In the case of a `TBrowseDB` object we can accept the default cursor movement methods created for us when we re-instantiate the object for the DBF concerned - we do not need to save them in the array. The code block that determines the contents of each column is another matter; we do need to save a stringified version of this, for which we can use the cargo slot in the `TBcolumn` in question.

It is now possible to write a pair of conversion functions that map a `TBrowse` object to an array and vice-versa. The array produced contains only simple data types and can be safely saved and restored to and from a disk file.

User configurable TBrowse

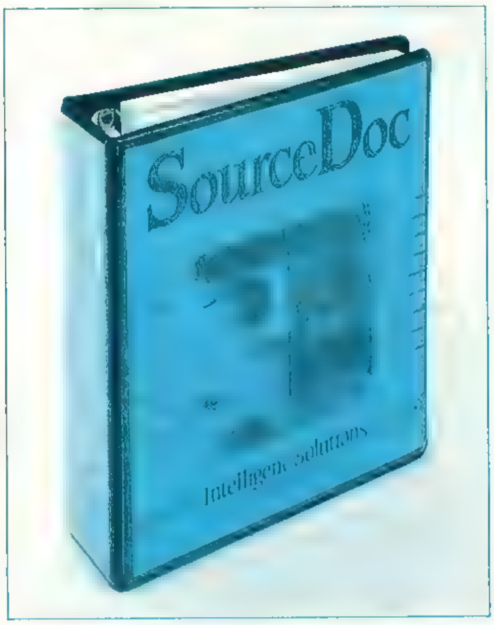
With the ability to store and retrieve `TBrowse` objects off-line instead of having to write code to recreate them afresh each time, we can realistically tackle a program to let users configure and save their own browse tables. With a little extra effort we might even be able to extend the concept to provide a powerful development tool - after all, methods can be implemented using stringified blocks, strings can be entered with a text editor and the syntax rules for code blocks are fairly straightforward...

The key handler methods array can be extended to include a second dimension for function keys, which can evaluate code blocks that manipulate the browse table itself. For example, a function can be written to allow the user to re-size and/or move the browse table on the screen. The function can be 'attached' to a function key by assigning an appropriate stringified block to the correct element in the methods array. As the user moves the table the appropriate instance variables are updated so that when the browse is saved the new co-ordinates are saved with it.

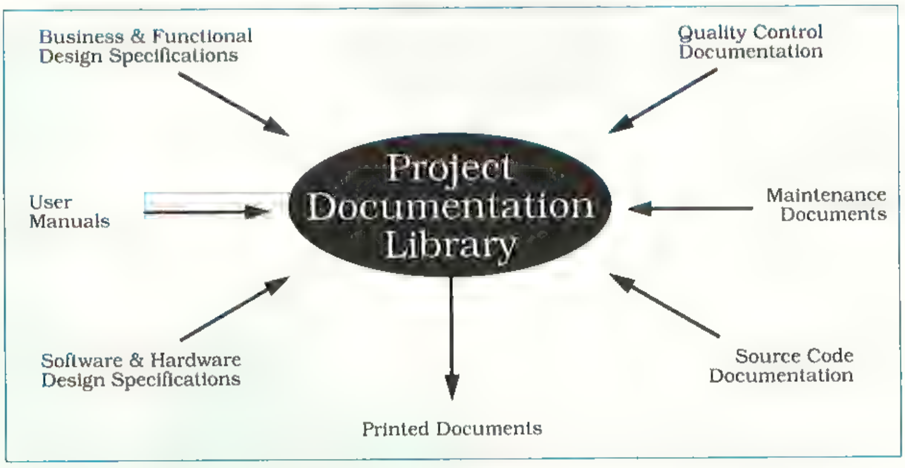
By building on these concepts, a dynamically configurable browse table can be constructed that allows the user to change almost anything, including the number, contents, titles, formatting and order of columns. With a little extra work an editor for changing stringified blocks can be devised and the key handler methods array updated on-line.

REVOLUTIONISE YOUR DOCUMENTATION WITH

SourceDoc™



- SourceDoc: The language-independent, fully automated HyperText documentation tool for professional programmers and software development groups
- SourceDoc creates a Dynamic Access Project Documentation Library for system documentation
- SourceDoc is currently used by the following companies: Ashton-Tate • Compaq • Epson • Fox Software • Hayes • Hewlett Packard • IBM • Intel • Lotus • Novell • SBT Corporation • Tektron IX • Texas Instruments • Xerox


Business & Functional Design Specifications
Quality Control Documentation
User Manuals
Project Documentation Library
Maintenance Documents
Software & Hardware Design Specifications
Source Code Documentation
Printed Documents

For further information call

Readmar Systems (071) 625 5255

APPROVED DEALERS: **Grey Matter Limited**, Prigg Meadow, Ashburton, Devon TQ13 7DF, Tel: (0364) 53499 • **Roundhill Computer Systems Limited**, Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU, Tel: (0672) 84535 • **System Science**, 3-5 Cynthia Street, London N1 9JF, Tel: (071) 833 1022 • **Reflex Technology Limited**, 9 Buckingham Place, Bellfield Road, High Wycombe, Berks HP13 5HW. Tel: (0494) 465907

CIRCLE NO. 446

Conclusion

Familiarity with all of Clipper 5.01's features and a modular approach to development can lead to a very high degree of reusability.

The concepts behind the code presented here can be applied to construct a set of tools that the developer, or even the end-user, can use to build an application. Applications built in this manner are easily maintainable and can be enhanced or extended easily. In addition, the quality of a system built out of tried and tested components is much higher, reducing maintenance overheads. The price, of course, is increased up-front development. There is often an overhead involved in modular design; do you hard code the report or write an *ad-hoc* report generator? Commercial reality usually prevents increasing the cost of the current development to save on future systems, and code is continuously rewritten from system to system.

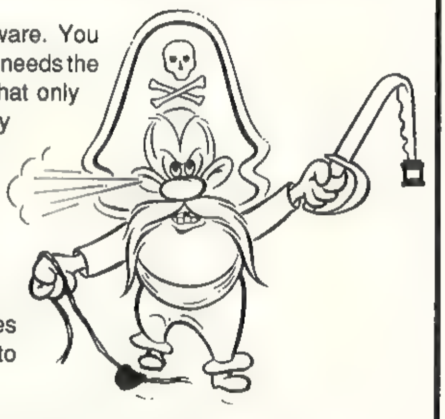
Perhaps the greatest single advantage of OOP is that it forces the developer to adopt a modular approach, but you have the option to adopt this approach anyway, you don't have to wait for OOP.

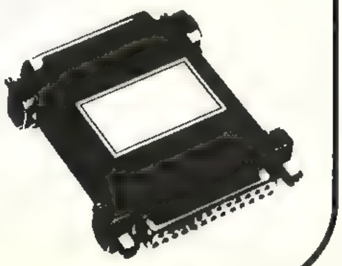
EXE

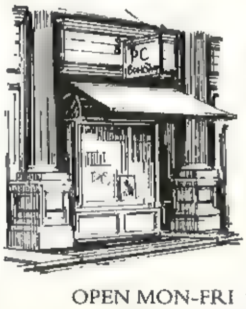
```
#include "inkey.ch"
#include "box.ch"

// tbo:cargo array contents:
#define TB_DBF   1  // .DBF File name
#define TB_NTX   2  // .NTX File name
#define TB_BOX   3  // Box characters
#define TB_METH  4  // Methods array
#define TB_SLOT  5  // Free slot

function aNewTbo(cDbf, cNtx)

 local i,;            // Array index
       oTb,;         // Tbrowse object
       aMethods,;     // Methods array
       cField,;       // Field name
       nKp            // Key pressed

 use (cDbf) index (cNtx)
 oTb := tbrowsedb(02, 10, 23, 70)

// Initialise methods array:
 aMethods := methods()

// Set up cargo slot to save info:
 oTb:cargo := array(TB_SLOT)
 oTb:cargo[TB FILE]  := cDbf
 oTb:cargo[TB_INDEX] := cNtx
 oTb:cargo[TB BOX]   : = B_SINGLE
 oTb:cargo[TB_METH]  := aMethods
 oTb:cargo[TB_SLOT]  := NIL

 for i := 1 to fcount()
   cField := fieldname(i)
    oTb:addcolumn(tbcolumnnew(cField,;
                     fieldblock(cField)))
  next

  cls
  @ 01, 09, 24, 71 box oTb:cargo[TB_BOX]
  do while (nKp := inkey()) <> K_ESC
    do while ((! oTB:stabilize())    ;
       .and. (nKp := inkey()) == 0)
    enddo
    if nKp < len(aMethods) .and. nKp > 0
      if valtype(aMethods[nKp]) == "C"
        eval(&(aMethods[nKp]), oTB)
      endif
    endif
  enddo

return NIL

function methods()

// Key array is of stringified blocks
// so it can be saved to a disk file:

  local aMet[40]

  aMet[K RIGHT]  := "{|b| b:right()}"
  aMet[K_LEFT]   := "{|b| b:left()}"
  aMet[K_UP]     := "{|b| b:up()}"
  aMet[K_DOWN]   := "{|b| b:down()}"
  aMet[K_PGDN]   := "{|b| b:pagedown()}"
  aMet[K_PGUP]   := "{|b| b:pageup()}"

return aMet
```

Figure 4 - Key handler with methods array

James Ormrod is a founder member of Applications Technology Ltd, a Nantucket Authorised Training and Development Centre. He is actively developing in 5.01 and wishes somebody would properly document the C and Assembler Extend System. ApT can be contacted on (0491) 26060.







Nantucket Tools II

The Clipper world is particularly rich in third party add-ons. However, Bob Rimmington has been looking at a new library from Nantucket itself.

The main contenders in the xBASE market have their particular merits and their protagonists. However it is not always sufficient to compare basic packages, as the extent to which a system can be supplemented and enhanced with supporting software may be more relevant. Clipper is especially strong in this respect, with a wide range of add-on packages. Now Nantucket, the company behind Clipper, has itself released a much extended toolkit function library.

Over 560 functions are provided. The types and numbers per category are set out in Figure 1. The toolkit is aimed squarely at Clipper v5.01, although most functions are said to work with the Summer '87 version as well. They are all included in a single .LIB file though some categories also require a supplied Extended Driver .OBJ file to be linked in. Various documentation, examples, font designer and other supporting files are also included. Installation is similar to Clipper itself, decompressing from the three (720 KB) distribution floppies onto the hard disc.

New Categories

Four categories are completely new to Tools II. For networks, just seven functions apply to PC LAN/MS-NET products, the remaining 37 to Novell. All of the latter support Netware 2.1 upwards and ELS II but not necessarily earlier versions or ELS I. The possible implementations are clearly indicated for each function. If you know your Novell you can access it via the Clipper RUN command, but Tools II not only makes things much easier but allows return values to be used and tested. LOGIN information such as IDs, rights and privileges can be obtained. Files can be copied and purged, the maximum and the free capacity of a Volume can be pre-checked in respect of both files and space. For printing, the Novell CAPTURE mode can be set, tested, activated and deactivated. Just one example, `NNETCAPSSF(1, FALSE, TRUE, 0, 5, 8, 2)` will specify device LPT1, set an end-of-run form feed, print banner page, select server printer No. 0, set a five second time out, set tab width to 8 and print two copies.

The second new category, maths, contains a mixture of trigonometric and financial functions. A third, Peek/Poke, enables direct reading and writing of bytes and 16-bit words to memory and designated ports. It is rightly accompanied by dire warnings of use by anyone who does not fully understand the possible consequences.

Tools II was originally written in German, and the translation to English has resulted in a few quirks. Not the grammar or style of any text, that is fine, but rather the occasional nonsense. For example, the introduction to the fourth new category, Get/Read functions, says it has been 'retained' for the sake of compatibility. `GETFIELDCOL()` and `GETFIELDROW()` return the column and row start positions of a `GET`, useful if you need to write to that position from a VALID UDF function. There is a `GETSECRET()` function that substitutes asterisks (*) for characters actually entered. Finally four functions for saving current `GET`s and `SET KEY..TO` settings to arrays and for restoring them. These enable new `GET` masks or key settings to be activated without losing the current ones.

Windows and Extended Drivers

The category that will first attract most new users is that for Windows. Let us get one thing straight from the start, this has nothing at all to do with Microsoft Windows. Instead it is a facility to place normal Clipper input/output in screen 'Windows' that can be moved, re-sized and overlapped by either the programmer or the user. For the latter, SCROLL LOCK must be activated and then the cursor keys act as one might expect. There is no Mouse option though this may be possible through one of the third party add-ons.

At the simplest level, the windows functions do little more than could be achieved by saving a screen area, writing to it and then restoring. However they do automatically replace both the content and colour attributes of all areas overwritten. The first plus comes with the positioning of output with `@ SAY` and similar statements. The coördinates apply to the window, **not** the screen. So a window can be repositioned to suit a client with no more than a change to the `WOPEN()` parameters. The second comes with the ability to alternate between up to 255 windows (memory permitting) and to write to the currently selected window.

Extended Driver functions in part support Windows operations, in part act on a normal screen. `DSETWINDOW()` enables the output of any external programs, such as DOS, to be confined to the current window. It is possible to address a virtual screen of up to 255 columns or 255 rows. There is a 32 KB memory limit, so a typical maximum combination would be 255 columns and

FUNCTION CATEGORIES AND NUMBERS	
Windows	22
Extended Drivers	51
Serial Communications	40
String Manipulations	88
Peek/Poke	9
Number/Bit Manipulations	28
Video	60
Disk Utilities	50
Printer	15
Date and Time	27
Miscellaneous	22
Database	7
Set Status	44
Network	44

Figure 1 - Functions by Category

Q.

"What will happen to dBASE?"

"When can I compile my dBASE code?"

"Will I have to learn Paradox?"

"How can I learn OOP at my own pace?"

"What will the next version of dBASE be like?"

"Will I have to learn another language?"

"Where's the professional compiler?"

"X" what?"

"Will there be a next version of dBASE?"

nantucket

Nantucket UK Ltd, 8 Bridgegate Centre, Welwyn Garden City, Herts. AL7 1JG. Nantucket logo and Clipper are trademarks of the Nantucket Corporation. Other brand and product names are used for identification purposes only and may be trademarks or registered trademarks of their respective owners.

CIRCLE NO. 449

128 rows. The screen then becomes a view into a far larger area. The size of a virtual screen is defined with SETMAXCOL() and SETMAXROW() functions. For xBASE programmers, after years of working to a limit of 25 rows and 80 columns, it seems uncanny to be able to put @ 111,222 SAY ... etc into a program. It does work though. An immediate use for this is where a normal entry screen is just not able to accommodate all the required GETs but paging to second or third screens is not ideal. The GETs just scroll up as those at the foot of the screen are entered, the up-cursor will bring back those at the top. Simple to program, simple to use. See Figure 2.

DSETQFILE() enables the writing of a 'Quit' file, comprising the current keyboard buffer, on any form of exit. An ideal way of checking for a normal exit or tracing the cause of a program failure. EGA43(), VGA(28) and VGA(50) allow alternative screen modes. GETKXCLAT(), GETKXTAB(), INKEYTRAP(), SETKXLAT(), SETKXTAB(), TRAPANYKEY() and TRAPSHIFT() allow sophisticated key trapping and action. INKEY() will never be the same again. KEYSEND() and KEYREAD() extend the use of the keyboard buffer.

External Devices

The Serial Communications category will not excite every Clipper developer but could prove invaluable to some. Among new functions for Tools II are XMOBLOCK() and XMOCHECK(), to generate and test X-MODEM blocks. For printing there are functions to check ports and printers, print the screen (very handy when logging application errors) and make various use of the DOS PRINT facility.

Many applications require access to drives or to do more than USE and INDEX files within the normal xBASE language. Such activities work through DOS and so make the application vulnerable to normally untrappable DOS errors. This can be a particular problem where the operator has to use a floppy drive for backup or similar tasks. The scope of the Disk Utilities category is now so wide that it is difficult to think of a circumstance it does not cover. DISKREADY() will check that a drive contains a disk and is ready to use. DISKREADYW() will check that it is not write protected. DISKFORMAT() will allow, through UDFs, both easier selection of format modes and control of format track repeats. Other functions allow interrogation and control of available disk space in relation to file sizes.

A complete sequential backup using a set of disks is possible with safety checks at every stage. FILESEEK() can be called with parameters to find the first matching file, then repeatedly called without parameters to find each subsequent file. Any developer who has found the combination of users and floppy disks a potential hazard would find this category alone well worth the money.

```
/* NANTUCKET TOOLS II  -  EXAMPLE CODE

   The code below illustrates Tools functions for
   Windows and for a Virtual Screen.  Both require the
   Extended Driver file NT2US50.OBJ to be linked in.
*/

#include 'INKEY.CH'              //... Include constants for key
codes
#define TRUE    .T.
#define FALSE   .F.

LOCAL nWinNum1,nWinNum2,nWinNum3,nKey:=0,nCtr

/* Open three Windows.  Write text to them.  Select,
   move and close Windows.  Pause using INKEY(0)
*/

CLS
@ 12,30 SAY 'This is normal screen'
*
WSETSHADOW('W/BG')               //... Transparent shadows
nWinNum1=WOPEN(5,5,15,50)        //... Open first window
WBOX()                           //... Draw default box round it
@ 2,2 SAY 'This is Window No. 1'
INKEY(0)
nWinNum2=WOPEN(10,25,20,60)      //... Open second window
WBOX()
@ 2,2 SAY 'This is Window No. 2'
INKEY(0)
nWinNum3=WOPEN(15,35,22,75)      //... Open third window
WBOX()
/.... Position text relative to border
@ WFROW(TRUE)+1,WFCOL(TRUE)+1 SAY 'This is Window No. 3'
INKEY(0)
//... Select first window, this brings it to the top
WSELECT(nWinNum1)
INKEY(0)
WMOVE(WROW()-3,WCOL()-3)         //... Move up and to left
@ 4,2 SAY 'Now moved towards top left'
INKEY(0)
WCENTER(TRUE)                    //... Move to centre of screen
@ 6,2 SAY 'Now centred on screen'
INKEY(0)
*
WCLOSE()                         //... Close current window (1)
INKEY(0)
WCLOSE()                         //... Close current window (3)
INKEY(0)
WCLOSE()                         //... Close remaining window (2)
INKEY(0)

/* Now back to normal screen with original text restored */

/* Define virtual screen, write text to right of and below
   physical screen.  Move about virtual screen with cursor keys
   in conjunction with INKEY(), DO WHILE loop and DO CASE test.
   If GETs are used, normal navigation with cursor keys causes
   physical screen to show relevant portion of virtual screen
*/

SETMAXCOL(121)                   //... Set number of columns
SETMAXROW(51)                    //... Set number of rows
*
@ 0,0 CLEAR TO 51,121            //... Clear (text etc. may remain
                                 //...   in unseen screen area)
@ 3,1 SAY 'Left of screen'       //... Write across virtual screen
@ 3,35 SAY 'From column 35'
@ 3,65 SAY 'From column 65'
@ 3,99 SAY 'From column 99'
*
FOR nCtr=5 TO 51 STEP 2          //... Write down virtual screen
  @ nCtr,5 SAY 'This is row '+NTOC(nCtr)
NEXT
*
@ 0,0 SAY ' '                    //... Return to top left
*
DO WHILE nKey !=27               //... Loop until Esc pressed
  *
  nKey:=INKEY(0)                 //... Read cursor keys with
INKEY()
  *
  DO CASE
  CASE nKey=K_UP
    FIRSTROW(FIRSTROW()-1)       //... Reset first row of screen
  CASE nKey=K_DOWN
    FIRSTROW(FIRSTROW()+1)
  CASE nKey=K_PGUP
    FIRSTROW(FIRSTROW()-10)
  CASE nKey=K_PGDN
    FIRSTROW(FIRSTROW()+10)
  CASE nKey=K_LEFT
    FIRSTCOL(FIRSTCOL()-1)       //... Reset first column of screen
  CASE nKey=K_RIGHT
    FIRSTCOL(FIRSTCOL()+1)
  CASE nKey=K_CTRL_LEFT
    FIRSTCOL(FIRSTCOL()-10)
  CASE nKey=K_CTRL_RIGHT
    FIRSTCOL(FIRSTCOL()+10)
  CASE nKey=K_ESC
    EXIT
  ENDCASE
  *
ENDDO
*
@ 0,0 CLEAR TO 51,121            //... Erase everything
FIRSTCOL(0)                      //... Reset for normal physical
screen
FIRSTROW(0)
*
RETURN

/*   EOF Demo program   */
```

Figure 2 - Example code for Windows and Virtual Screens

How much longer can you afford to wait?

Create Overlaid Programs-Fast.

BLINKER™, the world's first and fastest dynamic overlay linker, reduces your link time to seconds and reduces program memory requirements. Now you can use one linker for all your software projects.

One Linker, Many Languages.

BLINKER 2.0 links and automatically overlays DOS programs written in Microsoft® C, BASIC, Assembler, QuickBASIC™, Fortran, Pascal, Watcom™ C, Zortech C++, Clipper®, FORCE® and in Borland® C, C++, Assembler, and more.

Save Time and Memory.

BLINKER removes the need for overlay structures, simplifies program design and reduces memory requirements to save you time, effort and memory.

Memory Swap Function.

BLINKER is the ONLY linker to offer an integrated memory swap function, so you can run other large programs from within your program, with negligible memory overhead.

BLINKER™
High Performance Dynamic Overlay Linker

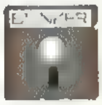
Don't Settle for Less.

Other major features include full CodeView® support, use of EMS/XMS at program run-time, and enhanced execution speed of overlaid code.

Time is Money.

BLINKER offers all this in a fraction of the time it takes to link with your current overlay linker. You know time is money, and link time is no exception.

Free Demo



To try our free demo on your own code and for more information contact our U.K. distributor:

Call: +44-81-994-4842 or
FAX: +44-81-994-3441

QBS Software Ltd.
10, Barley Mow Passage
London W4 4PH

BLINKER is available in 5.25" or 3.5" diskette format

Price £199
plus shipping & handling

VISA MasterCard

Blink*inc*
P.O. Box 7154
Richmond VA
23221

© 1991 Blinkinc. Blinker is a trademark of Assembler Software Manufacturers Inc.

CIRCLE NO. 451

Strings and Numbers

Attempting to illustrate the many string functions risks becoming just a boring catalogue. Some seem quite obscure at first, then a need arises and their value is appreciated. `REMALL()`, `REMLEFT()` and `REMRIGHT()` are like `ALLTRIM()`, `LTRIM()` and `RTRIM()` except that instead of just blanks any specified characters can be removed. Ever needed to remove trailing '0's from a number? Try

```
REMRIGHT(cMemvar,'0')
```

Many other functions allow selection and manipulation of characters within strings. `CHARSORT()` allows sets of characters to be sorted within a string with parameters to set the length of the sorting element, offsets and other options.

Three functions enhance the use of others. `CSETREF()`, if called with `TRUE`, suppresses a return value. When used with a long string passed by reference it avoids wastage of memory on a duplicate string. `CSETATMUPA()`, if called with `TRUE`, sets a multi-pass mode on for functions such as `ATREPL()` (search for a sequence within a string and replace it) and `NUMAT()` (count the occurrences of a sequence in a string). `SETATLIKE()` enables wildcards to be used in `AT...()` and similar functions.

A group of 'Token' functions can be used to identify words, sentences, dates and other sets of characters within a string. A 'Token' is a sequence of characters within delimiters. These could be spaces and all the normal punctuation symbols to identify words. In this context a word could have a different delimiter at each end such as a space before and a comma after). Typical uses for the token functions include neat word-wrapping of long text fields in a report and fast text searches. For the latter, words in a text field could be extracted, saved (with a record ID) in a separate indexed file and then found with the `SEEK` command.

The Number/Bit category allows conversion of numbers to a different base and various binary operations such as `AND`, `OR`, `XOR` and `NOT`. Other functions include two for random numbers and another to convert floating point numbers to 8-byte strings.

Other Categories

A further 184 functions are in the six categories not already mentioned. Space permits mention of only a few samples. `NTOCDOW()` and `NTOCMONTH()` change week and month numbers into names. `SHOWKEY()` enables screen display of the status of the NUM-LOCK, SCROLL-LOCK, CAPS-LOCK and INSERT keys. `ENVPARAM()` reads the DOS environment into a string.

If you find the combination of users and floppy disks a hazard, then these functions are well worth the money

The Video category include alternate fonts and palettes while Miscellaneous includes sound. Some example files are supplied for the latter. For the client who has everything, how about a message in Cyrillic script as the screen slowly fades from one shade to another? Or a computer rendition of the Moonlight Sonata in the background? Perhaps not but, if he does insist, it is now possible. `SCREENWRITE()` writes a screen to a file, `FILESCREEN()` will read it back again. Functions such as `CLEARSLOW()` and `SAYMOVEIN()` allow visual effects, similarly `GETBOXGROW()` for a controlled box explosion.

Documentation

The toolkit in part extends the scope of Clipper itself, in part provides an off-the-shelf and faster alternative to routines a programmer could write for himself. If the potential saving in development time is to be realised good documentation is essential. Nantucket scores high marks with its main reference material. Every function is fully described, with brief examples, in a set of three clear, ring-bound manuals. The same information is repeated in a data file for use with the Norton Engine supplied with Clipper itself. The latter is good providing you know the relevant category, as this must be selected first. It is a pity that no overall search facility is included for those occasions where the precise syntax for a known function name is required.

Unusually, what seems to be the same data is supplied in a third format in a text retrieval system. This will search for all entered strings, including wild cards. Although I commend Nantucket's intentions, I have to confess that I have found it of little value. It is not a TSR and so is not easily accessed while writing code. Nor does it tell me anything fresh. Sorry Nantucket, perhaps the penny will suddenly drop one day. A surprising omission is any instruction on how to start it though few will fail to try the obvious solution of just entering the name of the .EXE file.

Less satisfactory is the limited provision of 'how to use it all' guidance. There are some example program files, enough to prove how valuable these can be, too few to cover most functions. Query By Example is very much flavour of the month for users, why not Program By Example for developers? Many would find this more beneficial than three versions of the formal instructions. The example programs include only minimal and often inadequate comment. Not all use the current Nantucket convention for naming variables. Except for the examples, no source code is supplied.

Performance

All the functions tried worked without problem. Tools II includes a 'two screen' Debug facility for those with both VGA and Hercules cards in their machines and of course the corresponding monitors. It was not possible to try this (the /S switch used with CLD gives a comparable facility on one screen using the 50 line VGA mode) but with several functions calling the debugger did seem to cause some problems. Use of Blinker in place of RTLINK seemed ok, but only if the Tools II .LIB and .OBJ files were not placed in the overlay area.

Conclusion

Many of the Tools II functions open up areas not previously accessible to Clipper developers. With others, users will save much of the time spent building their own personal library. For the latter purpose alone saving just two days would make it self-financing. Few will find a need for all 565 functions. However it is the benefit from those that are used that matters. The Toolkit is comprehensive, well presented and, notwithstanding the small reservations, is well documented. It is the availability of additional products of this quality that help make Clipper the preferred choice of many xBASE developers.

EXE

Bob Rimmington is an independent xBASE and Clipper consultant. He can be contacted through Stanford Systems on 0444 236352. Nantucket Tools II is priced at £495. Nantucket is on 0707 373600.

Using Pre-Link Libraries

Interested in linking your applications more quickly and decreasing your disk space usage? Aric Rosenbaum explains how Clipper's PLLs can accomplish all of this and more.

With the introduction of Pocket Soft's .RTLink, a new method of linking has arrived on the Clipper scene. More specifically, we are talking about run time or pre-link libraries. Since Nantucket includes a copy of .RTLink with every copy of Clipper 5.0, this feature is available to every 5.0 user. This article will explain what pre-link libraries are and why they should be used. In addition, the usage and creation of pre-link libraries will be explored.

Background

Pre-link libraries (PLLs) provide a core of routines that are common to various different applications. They do not need to be compiled and linked every time an application is created. In addition, all of your Clipper programs can rely on one pre-link library. This significantly reduces link times, disk space, and modem transfer times.

For instance, it is almost a certainty that every one of your Clipper applications has a function that will centre a string on the screen. This is one of the many dozens of functions that developers think nothing about including in an executable. However, every time you create a new version of your program, you must compile and link the centre function. If you are truly organised and have placed the centre function in a library, you only need to deal with the time consuming linking. This is wasteful. Why not compile and link the function once? The code does not change. In addition, this code could be used as part of a common code bank for all your applications.

Pre-link libraries do just this. Instead of linking all you routines into a giant .EXE file (of the type for which Clipper is justly notorious), you create a smaller .EXE and supply a separate .PLL file. When your program is run, special code loads up the .PLL so that it can use the library functions. The reduction in disk space is achieved by having several applications share the same PLL.

Using .RTLink, you instruct the linker to create your own pre-link library. When linking an application, you inform the linker that you will be using a pre-link library. When the application is executed, the program will load and use the pre-link library.

Benefits

As I have hinted, the benefits of PLLs are substantial. They include decreased link times, decreased disk space and decreased modem transfer times. To illustrate the above three conclusions, two applications were tested. The first is a very short program that uses some of Clipper's features. This program, shown in Figure 1, utilises commands and features such as: `SKIP`, `eof()`, `SAY`, and `GET`. Using a one line program that says, 'Hello World' does not properly convey why Clipper is used. Since real life applications rarely work like test programs, I also compare a Summer '87 program that was recently converted to 5.0. In summary, linking times were as much as 4.8 times faster and applications were as much as 32.8 times smaller. Imagine a Clipper .EXE file that takes up as little as 7168 bytes of disk space! However, please note that I did not say that the executable was 7 KB. The executable still requires a 277 KB PLL.

If your clients are supported by modems and you routinely transfer new versions, these small executables are of interest. For example, instead of spending 16 minutes transferring an executable with a modem, you could spend 30 seconds. This enhances your ability to get your client back on-line as quickly as possible and keeps your money out of the telephone company's pockets.

Now, after all these fantastic claims, I am sure a few cynics exist. You might be saying, 'Oh sure. And do I get a free pair of Ginsui steak knives.' Well, I cannot promise the steak knives, but the claims are true. Figures 2 and 3 show the results for the small test program.

These are pretty impressive numbers. All of this with the pre-link library that ships with Clipper 5.0. This pre-link library includes CLIPPER.LIB, EXTEND.LIB, TERMINAL.LIB and DBFNTX.LIB. While the raw results for the real-life applications are not as awe-inspiring, they are still very credible. These results can be seen in Figures 4 and 5.

Again, the above numbers are based upon using Nantucket's BASE50.PLL pre-link library. By pre-linking the common libraries that I normally use, even better results can be obtained.

```
CLOSE DATABASES
CLEAR SCREEN
USE test_dbf INDEX test_ntx ALIAS test NEW
GO TOP

DO WHILE .not. eof()
   SKIP
ENDDO

sample := space(10)
@ 10,10 SAY 'Anything: ' GET sample VALID (.t.)
READ

? sample

RETURN
```

Figure 1 - Source code for small test program

Drawbacks

As everything else in this world, there are drawbacks to using pre-link libraries. These problems, however, are minimal. The first disadvantage is that using pre-link libraries creates a larger load module. The size difference varies with each application but is somewhere between 6 KB and 13 KB. With the introduction of VMM and dynamic overlays, this small overhead does not present a problem.

The second obstacle is more of a problem of logistics. What if the user, running Xtree after a heavy night on the glue, deletes the pre-link library? He will not only lose the application in the current directory but all the Clipper applications. This is regarded as the biggest detriment to pre-link libraries. For this reason, developers are leery of having multiple applications rely on one core. I, however, do not see this as a significant risk If you are really windy, marking the file as read-only

```
ATTRIB +r MY_PLL.PLL
```

will thwart all but the most determinedly suicidal users.

Using PLLs

Using pre-link libraries with your executable is so easy that I considered omitting this section. Nothing special needs to be added or changed in your executable. In addition, no special driver or TSR needs to be loaded before loading your application. When the linker links your application, it will properly inform the executable that it needs a pre-link library.

When executing an application that requires a pre-link library, the program will look in one of two places for the library. The first, of course, is the current directory. If a pre-link library is specific to a particular application or only one Clipper application exists in the system, the library will probably be placed in the same directory as the executable. However, if multiple applications are using the same pre-link library and these applications reside in distinct directories, a different approach needs to be taken. One approach is to place each application in its own directory and locate the pre-link library in a unique directory. When an application is executed it will load the pre-link library from the other directory. To do this, you have to tell Clipper where the pre-link library resides. This is done with an environment variable:

```
SET PLL=C:\PLL
```

The above environment variable will tell the application to search the PLL subdirectory on the C: drive if the proper pre-link library cannot be found in the current directory. If your pre-link libraries reside in different places, you specify the locations and separate them with a semicolon.

	w/ BASE50.PLL	w/o PLL
Size of TEST.EXE:	7,168	235,520
Size of BASE50.PLL:	277,504	n/a
Link time:	20.9 sec.	96.0 sec.
Transfer time with a 2400 baud modem:	30 sec.	984 sec.

Figure 2 - Raw results for the small test program

TEST.EXE	32.9 times smaller
Link time	4.8 times faster
2400 baud modem time	32.9 times faster (and less expensive)

Figure 3 - Comparative results for the small test program

	w/ BASE50.PLL	w/o PLL
Size of REAL.EXE:	105,472	343,040
Size of BASE50.PLL:	277,504	n/a
Link time:	67.1 sec.	159.1 sec.
Transfer time with a 2400 baud modem:	7.3 min.	23.8 min.

Figure 4 - Raw results for the real-life application

Linking with PLLs

If you are planning to use a pre-link library with your application, you must inform the linker and tell it what pre-link library will be used. To inform the linker that pre-link libraries will be used, a command is inserted in the link script file (ie .LNK file). If you are using the command line mode, you will need to use the '/PLL' switch. If this command is not included, the linker will either link the normal library or complain about undefined symbols. It depends on what is missing. If the pre-link library is not specified, the linker will automatically link CLIPPER.LIB, EXTEND.LIB, TERMINAL.LIB and DBFNTX.LIB. However, if the pre-link library is not specified and the required function is custom, the linker will issue an undefined symbols warning.

Specifying the pre-link library can be done in one of two ways. The first and most common is in the link script file. For example, when linking the test program, I used the following link script file:

```
FILE TEST
PLL BASE50
```

Alternatively, if I issued the link command from the DOS prompt:

```
RTLINK FILE TEST /PLL:BASE50
```

It really is no more difficult than including a normal library.

Creating PLLs

Creating pre-link libraries, while more difficult, is almost as easy as using them. In summary, it requires:

- Informing the linker that you are creating a pre-link library.
- Specifying the object files and libraries to search.
- Referencing the modules that are to be included.

The first step is accomplished by including the command word 'PRELINK' in your link script file. When the linker parses the script file, it will know that it should create a pre-link library instead of an executable.

The second step is the same as the normal linking process. Specify the object code files and libraries that should be used to construct the pre-link library. For example, if we are using the following to build our pre-link library:

Object files	Libraries
APP_OBJ1	COMM_LIB
APP_OBJ2	MOUS_LIB
APP_OBJ3	SCR_LIB

The link script file would look as follows:

```
PRELINK
OUTPUT MY_PLL

FILE APP_OBJ1, APP_OBJ2, APP_OBJ3
LIB COMM_LIB, MOUS_LIB, SCR_LIB
```

LOOK at all OPTIONS.

Software Protection - but which TYPE ?

- **EVERKEY II** — *The Hard ware Option?*
  SINGLE WIRE ZERO LOAD - Total Compatibility.
  Not available in ANY other Key product.
- **EVERLOCK** — *The Soft ware Option?*
  Software-Only Copy Protection; economical & SECURE.
- **EVERTRAK** — *Non Copy-Protection Option?*
  Non Copy-Protection - Anti Hacker software protection.

CALL FOR A FREE WORKING DEMO DISKETTE

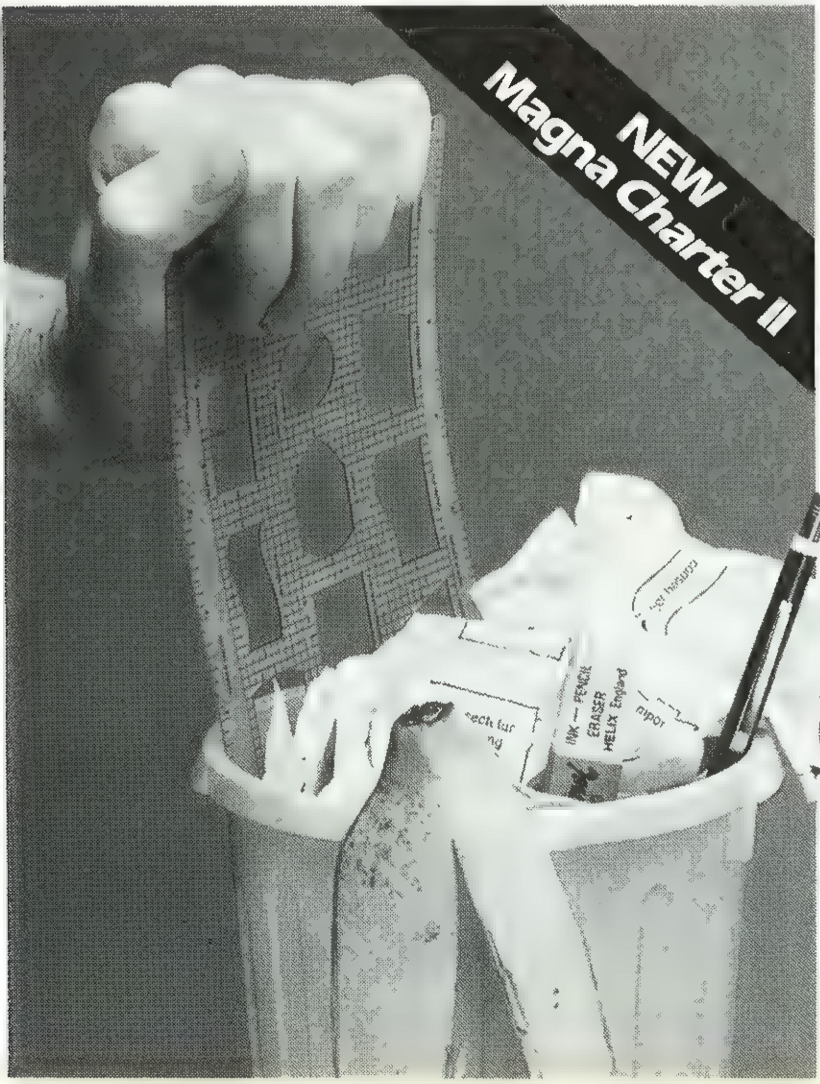
Other Services include:- Disk Duplication & supply - Label Production & supply - Packaging.

USER FRIENDLY
[illegible]
22A, Bartleet Road,
Washford Industrial Estate,
Redditch, Worcestershire,
B98 0DG. England.
Tel: 0527 510 105
Fax: 0527 514 229

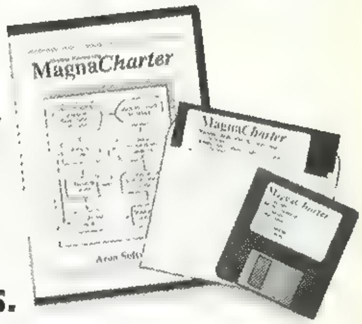
Sole Distributor for Az-Tech Software,inc - UK, Ireland & Spain

...... THE ONLY OPTION.

CIRCLE NO. 455



Out with the Old In with the New



Magna*Charter* II relieves the pain of drawing charts.

Magna*Charter* was one of the success stories of 1990. **Magna*Charter* II** is the latest version of this much acclaimed product — for Microsoft Windows 3. Based on the spreadsheet principle, **Magna*Charter* II** lets you build any kind of chart in minutes rather than hours and includes all the standard flowcharting symbols. Use the "crowsfeet" for database diagrams.

Magna*Charter* II divides the screen into cells, identified by a double mouse click or keyboard entry into which symbols are placed. All available symbols are displayed as icons — or add your own. The various features are accessed via drop down menus, dialogue boxes and multiple windows with a wide range of text styles and sizes available. Editing is by simple cut-and-paste techniques and links between symbols can be "auto" or "conditional". **Magna*Charter* II** supports PostScript output and a wide range of printers. Through the Clipboard, pass bitmaps, metafiles and text.

Magna*Charter* II supports Windows 3 (**Magna*Charter*** 1.03 is still available for DOS). Published by Systemstar, **Magna*Charter* II** is available for £160 excluding VAT and P&P. Price includes a complete manual and examples to help you achieve professional results fast.

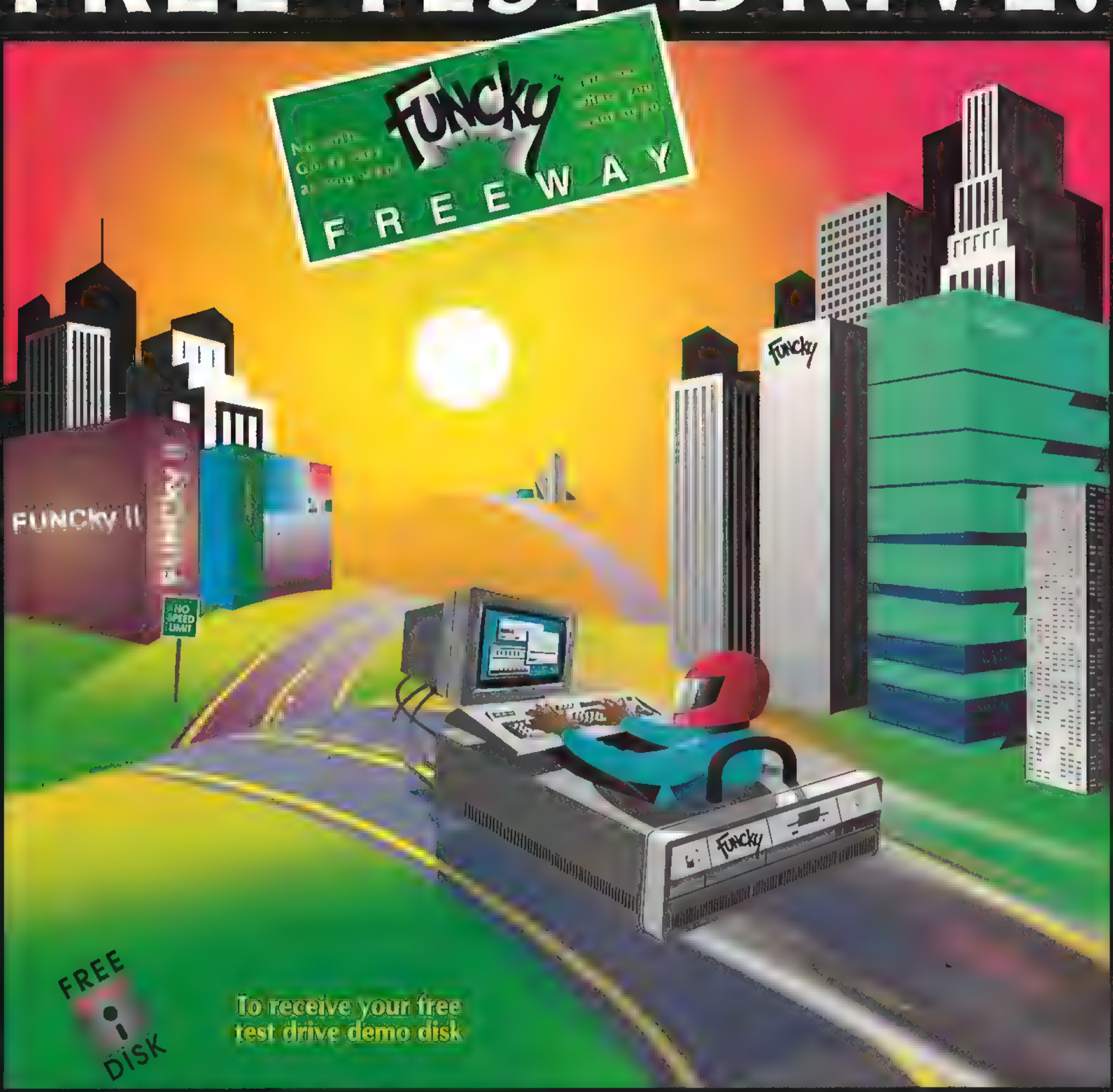
Order yours today by contacting Systemstar on (0992) 500919.

SYSTEMSTAR
LIMITED

1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 456

FREE TEST DRIVE!



Try our innovative TEST DRIVE program, a free demo of the FUNCky library which you can actually program with. Edit code using any or all of the FUNCky library, then compile, link and run to find out for yourself why FUNCky is the ultimate Clipper Developer's tool.

The FUNCky add-in library for Clipper contains over 450 functions which can be used as if they were a part of the Clipper language itself. Extensive features include moveable resizeable windows with mouse support, superior DOS file handling, blazing fast string manipulation, and impressive video control.

Get FUNCky now and get the phenomenal programming power you need to create the advanced applications demanded of you in today's competitive development environment.

dLESKO
INCORPORATED
320 York St., Jersey City, NJ 07302
FAX# 201-435-0637

Contact QBS Software

Telephone: +44 81 994 4842
Fax: +44 81 994 3441
BBS: +44 81 747 1979

CIRCLE NO. 457

It is important to note that the object code files that are listed contain common functions and procedures that we want to include in our pre-link library. You do not list all the object code files that make up your application.

An excellent way to illustrate the creation of a pre-link library is with a short analogy. Picture walking into a room full of books. You walk up to a stately oak table and sit down. Sitting in front of you is a completely empty table. Your project is to create a compilation of chapters from numerous books. Your first step of course is to get the books you need (object code files and libraries) and place them on the table in front of you. This done, you must now tell your assistant (linker) what chapters to include in the compilation. You do this by referring to the chapters (modules) in question. Once referenced, your assistant will dutifully copy the needed chapters into the compilation. After a minute or two (your assistant was a speedy, scholarly monk in a previous life), your compilation is complete and ready to be used by anyone who pleases.

Now that we have specified the object code files and libraries to search, we need to tell the linker what modules to place in the pre-link library. This is done with the `REFER` command. Following the `REFER` command are all the modules that should be included in the pre-link library. In our example COMM_LIB contains the following modules:

```
dial()
set_baud()
set_parity()
xfr_kermit()
xfr_x()
xfr_z()
```

We only want to include `dial()` and `set_baud()` in our pre-link library. We can do this by adding the following lines to our link script file:

```
REFER DIAL SET_BAUD
```

The link script file now looks as follows:

```
PRELINK
OUTPUT MY_PLL

FILE APP_OBJ1, APP_OBJ2, APP_OBJ3
LIB COMM_LIB, MOUS_LIB, SCR_LIB
REFER DIAL SET_BAUD
```

Based upon the above link script file, MY_PLL.PLL will contain two modules: `dial` and `set_baud`. Although we instructed it to search three object code files and two other libraries, only two modules were included. To include the modules in the other libraries, we would need to reference them.

One last note is that you should not worry about undefined symbols when creating a pre-link library. This is only a warning message. The undefined symbols will normally be resolved when the executable is linked. If, however, you receive an undefined symbols warning when linking the application then it is possible that the pre-link library was not properly created.

Real.exe	3.3 times smaller
Link time	2.4 times faster
2400 baud modem time	3.3 times faster

Figure 5 - Comparative results for the real-life application

```
PRELINK
OUTPUT BASE50

LIB CLIPPER, EXTEND, TERMINAL, DBFNTX
REFER _main
REFER _VOPS, _VMACRO, _VDB, _VDBF, _VDBFNTX
REFER _VTERM, _VPICT, _VGETSYS
REFER _VDBG
```

Figure 6 - BASE50.PLL link script file

```
PRELINK
OUTPUT MY_PLL

LIB CLIPPER, EXTEND, TERMINAL, DBFNTX
LIBRARY COMM_LIB MOUS_LIB SCR_LIB
FILE APP_OBJ1 APP_OBJ2 APP_OBJ3
REFER _main
REFER _VOPS, _VMACRO, _VDB, _VDBF, _VDBFNTX
REFER _VTERM, _VPICT, _VGETSYS
REFER _VDBG
REFER DIAL SET_BAUD
```

Figure 7 - Revised My_Pll link script file

Creating BASE50.PLL

As previously stated, Clipper 5.0 is shipped with BASE50.PLL, a compilation of CLIPPER.LIB, EXTEND.LIB, TERMINAL.LIB and DBFNTX.LIB. If creating your own pre-link libraries is of no interest, you should, at a minimum, use BASE50.PLL. The code in this library will not, of course, change and you will experience quicker linking times. If you would like to re-create BASE50.PLL, you can use the link script file that appears in Figure 6.

A few important points should be made. First, the capitalisation of the references is critical. As explained, `REFER` is responsible for pulling in certain modules. Changing lower-case to upper-case or vice versa will affect what modules are included in BASE50.PLL.

Second, if you want to create your own pre-link library that includes the above libraries, simply modify the link script file. All you need to do is change the output file name, include the appropriate libraries and object code files, and insert the correct references. For example, if I wanted to include the Clipper libraries in MY_PLL.PLL, I would use the link script file that appears in Figure 7.

Conclusion

Pre-link libraries are a great benefit to anyone who develops with Clipper. They decrease link times, disk space, and modem transfer times. These three benefits far outweigh the two minor drawbacks of a slightly increased load size and modest increase in vulnerability to dopey users. After experimenting with pre-link libraries, I am pleasantly surprised and plan to incorporate them into all of my future 5.0 applications.

Aric Rosenbaum is President of AR Consulting, Inc. His firm, located near New York City, specialises in the creation of custom software for a growing list of local and global clients. Aric is very interested in your comments and can be reached in the US on 0101 201 301 0909 (telephone), 0101 201 301 0128 (fax) or on CompuServe as 72460,1177.

Great Expectations

Le Compiler Nouveau est arrivé! Paul Kemp has been poring over a pre-release copy of Microsoft C/C++ version 7.0 - the world's most long-awaited compiler.

No one can accuse Microsoft of a headlong rush to OOP. The compiler giant has been quietly biding its time while its competitors push out new versions of their C/C++ compilers with evangelical zeal. Microsoft C/C++ 7.0 (C/C++7) has been a long time coming but it represents a colossal upgrade. To merely bump up the version number from 6.0 to 7.0 seems almost self-deprecating on the part of the company. I must stress that this article is very much a 'first impressions' overview of the product. There are two reasons for this: One, at the time of writing, the product was still at a beta V5.0 (called 'Pete' after Pete Best the Beatle - previous betas were named 'John', 'Paul', 'George' and 'Ringo'). It would therefore be unwise and unfair to benchmark the compiler and IDE properly. Two, we plan to do an in-depth analysis of the product in a future issue of *.EXE*. This will probably be when the next release of the compiler (probably v7.1) is available.

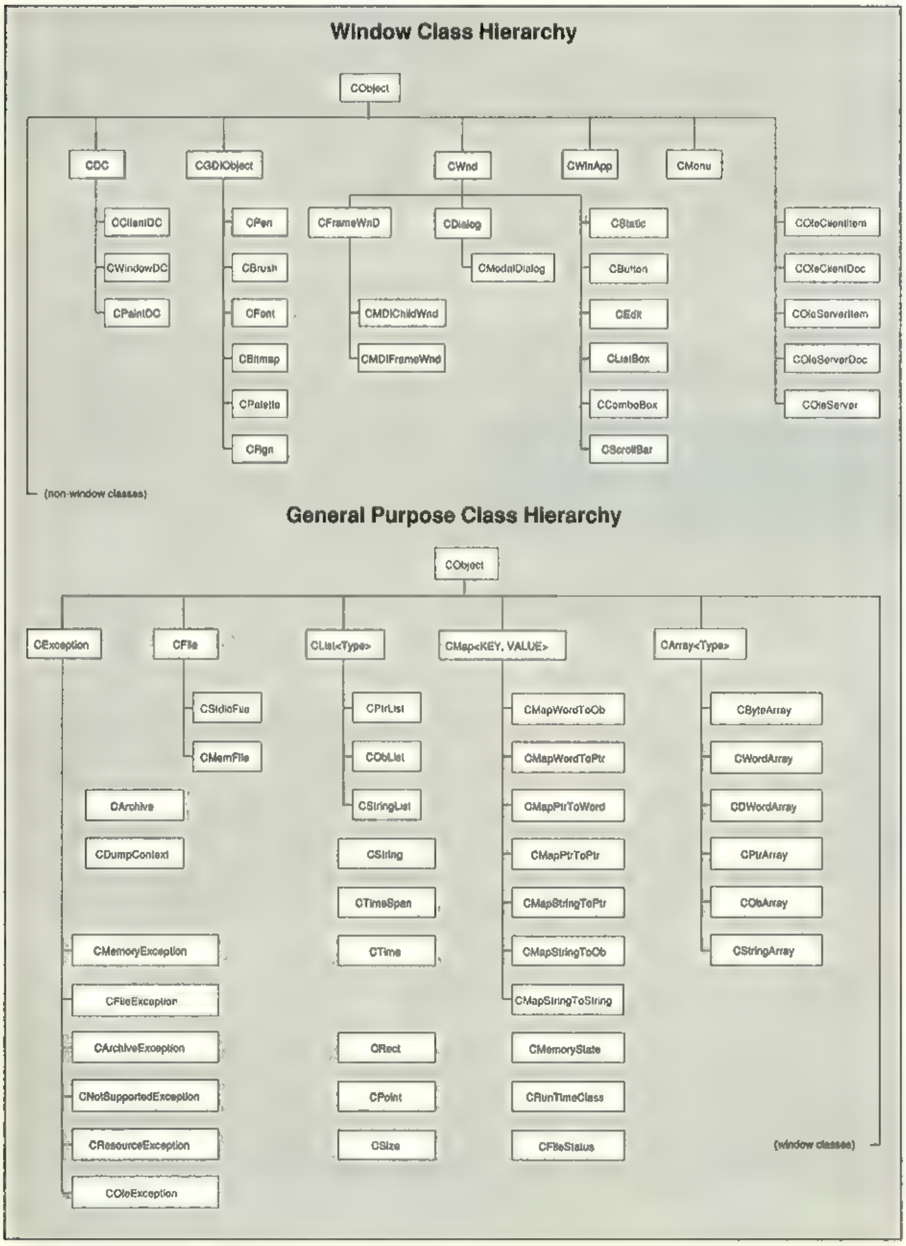


Figure 1 - Microsoft Foundation Classes (MFC) hierarchy

The basics

C/C++7 is a 32-bit protected mode C and C++ (v2.1) compiler, although as yet it does not support the creation of 32-bit DOS or Windows applications (this will be rectified in the next release). Microsoft has dropped support for low-end development machines altogether and the product requires an 80386 processor (or higher) and at least 4 MB RAM. The compiler, linker and PWB are DPMI-hosted under DOS, or can be run in a DOS box under Windows. In another controversial move, all support for OS/2 has been withdrawn. Such action is surely counter-productive when other software vendors are breaking their backs to provide cross-platform support for their products. However, Microsoft UK Head of Languages, Andrew King, did indicate that there is a good chance of the compiler supporting OS/2 v2.0 in a future incarnation.

The basic Windows SDK has disappeared (at last) and all the necessary tools for developing Windows apps (Dialog Editor, Font Editor, Bitmap Editor etc) are 'in the box' along with the resource compiler, header files and libraries. Specialist SDKs, for things like multimedia and pen extensions, still have to be purchased separately.

CodeView, now up to version 4.0, has been rewritten from scratch and implements a host of improvements, including full sup-

THE SOFTWARE CONSTRUCTION COMPANY

UPGRADE TO MICROSOFT C/C++ 7.0

Available directly from THE SOFTWARE CONSTRUCTION COMPANY LTD. your upgrade to Microsoft C/C++ 7.0 for only £149 plus V.A.T., includes next working day courier.

Competitive Upgrades to Microsoft C/C++ 7.0*

Topspeed C++	£149
Turbo/Borland C++ (any version)	£149
Zortech C++ (any version)	£149

** Requires copy of invoice to prove that competing product was purchased prior to February 1st 1992.*

How do I upgrade from Microsoft C 5.1/6.0 ?

To upgrade from Microsoft C 5.1/6.0 to Microsoft C/C++ 7.0 please supply a copy of your original invoice or the set-up diskette, together with a cheque for £175.08 (£149 plus V.A.T.) or credit card details, Access/Mastercard or Visa accepted.

How do I upgrade from a competing C++ compiler ?

To upgrade from either Borland or Turbo C++, Topspeed C++ or Zortech C++, which must have been purchased prior to 1st February 1992, please supply a copy of your original invoice, together with a cheque for £175.08 (£149 plus V.A.T.) or credit card details, Access/Mastercard or Visa accepted.

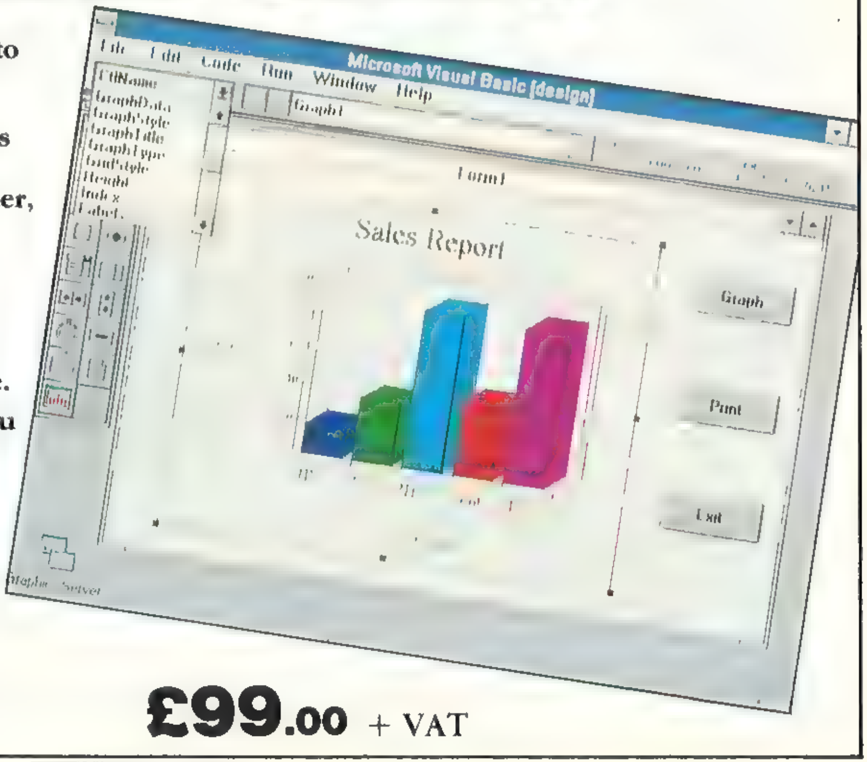
CIRCLE NO. 458

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

CHARTBUILDER

FOR VISUAL BASIC

- Integrate graphs and charts directly into your Visual Basic applications.
- Manipulate your graph objects using standard Visual Basic properties, events and methods.
- Present your data as pie, bar, line, scatter, area, gantt, polar and high-low-close graphs.
- Add statistical lines and trends.
- Create high-quality hardcopy.
- Interchange images with other applications via Clipboard and Metafile.
- Pay no royalties on the applications you create.
- ChartBuilder comes complete with a royalty free run-time copy of Bits Per Second's Graphic Server, the unique charting library for Windows developers.



Bits Per Second Ltd
14 Regent Hill, Brighton BN1 3ED
Tel: (0273) 727119 Fax: (0273) 731925

£99.00 + VAT

CIRCLE NO. 459

port for C++, remote debugging (via RS232 serial link) and DOS-extended applications. The Programmer's WorkBench (PWB), much-criticised when it first appeared in v6.0, has also been revamped in version 2.0. A new overlay manager called MOVE (Microsoft Overlay Virtual Environment) is also included in the package to compete with Borland's VROOMM, and a copy of the company's Source Profiler - previously only available as a seperate product - has been thrown in for good measure.

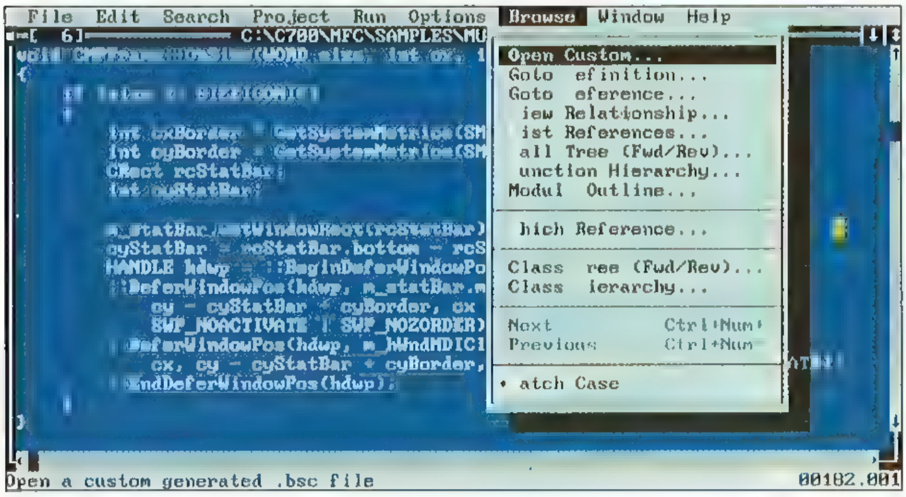
What's new?

Obviously the major new feature of C/C++7 is its support for C++, but it is also an upgrade of C 6.0 for straight C programmers. A number of new size and speed optimisations have been introduced including the facility to generate packed code or 'p-code'. This technology has been used internally at Microsoft for several years and is employed in some of its home-grown applications such as Word, Excel and PowerPoint. P-code can shrink the memory requirements of an application by 40 to 60 percent, although of course there is a run-time overhead. For DOS programs it may be used to compress code that might otherwise require overlays, and under Windows it is advantageous to have programs as small as possible to allow the operating system to juggle memory more effectively.

With p-code, application source code is compiled into 'interpreted object code', which is a higher-level and more compact representation of object code. A small (9 KB) interpreter module is then linked into the application so that no other special software needs to be loaded.

P-code generation is turned on by a simple compiler optimisation switch (/Oq) and requires little effort on the part of the programmer. A `#pragma` directive can also be used to generate p-code for selected non-time-critical functions in a single source file, so that p-code and native code can be included in the same object file.

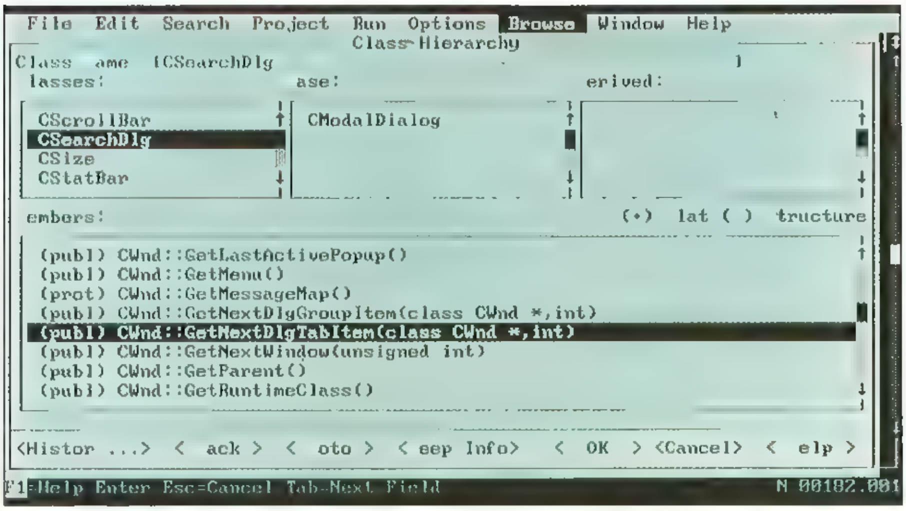
Microsoft recommends that user-interface code is a good candidate for p-code because under Windows the user perceives almost no decrease in speed (presumably because Windows itself is so slow).



```
v[illegible] (WORD size, int cx, i
{
    if [illegible] != SIZEICONIC)
    {
        int cxBorder = GetSystemMetrics(SM
        int cyBorder = GetSystemMetrics(SM
        CRect rcStatBar;
        int cyStatBar;

        m_statBar.GetWindowRect(rcStatBar)
        cyStatBar = rcStatBar.bottom - rcS
        HANDLE hdwp = ::BeginDeferWindowPo
        ::DeferWindowPos(hdwp, m_statBar.m
            cy - cyStatBar + cyBorder, cx
            SWP_NOACTIVATE | SWP_NOZORDER)
        ::DeferWindowPos(hdwp, m_hWndMDICl
            cx, cy - cyStatBar + cyBorder,
        ::EndDeferWindowPos(hdwp);
    }
}
```

Open a custom generated .bsc file 00182.001

Figure 2 - PWB's Browse menu


File Edit Search Project Run Options Browse Window Help
Class Hierarchy
<Histor ...> < ack > < oto > < eep Info> < OK > <Cancel> < elp >

```
Class ame [CSearchDlg                    ]
lasses:                ase:                erived:
 CScrollBar            CModalDialog
 CSearchDlg
 CSize
 CStatBar
embers:                            (•) lat ( ) tructure
 (publ) CWnd::GetLastActivePopup()
 (publ) CWnd::GetMenu()
 (prot) CWnd::GetMessageMap()
 (publ) CWnd::GetNextDlgGroupItem(class CWnd *,int)
 (publ) CWnd::GetNextDlgTabItem(class CWnd *,int)
 (publ) CWnd::GetNextWindow(unsigned int)
 (publ) CWnd::GetParent()
 (publ) CWnd::GetRuntimeClass()
```

F1=Help Enter Esc=Cancel Tab=Next Field N 00182.001

ADVANCED BUSINESS GRAPHICS

Figure 3 - PWB's class hierarchy browser

Enhanced speed optimisations include unrestricted function inlining, automatic function inlining and better near data allocation.

Microsoft also claims to have dramatically improved compilation speeds with C/C++7. I must confess that on the admittedly trivial applications that I tested, I did not find a significant improvement over C 6.0. However, I shall reserve judgement until a retail version of the compiler can be tested. Taking the cue from Borland C++, Microsoft has now implemented pre-compiled headers. The use of these obviously can reduce compilation times greatly. Microsoft's implementation has the advantage over Borland's of allowing executable code to be pre-compiled as well as header information. An unfortunate side-effect of this flexibility is that it is more cumbersome to use than Borland's offering. Because literally anything can be in a pre-compiled header, the programmer has to tell the compiler where the 'header' stops. This is done by embedding a `#pragma hdrstop` in the source file (see Figure 5). A pre-compiled header file has to be explicitly created, maintained and used through a number of compiler switches.

One of the main criticisms levelled at MSC6 was its inadequate documentation. Microsoft has responded to this by supplying heaps of the stuff with version 7.0. The 10-manual set runs to about 5,000 pages and is pretty comprehensive. Most of the information is also available online in the PWB (although I'm not a fan of the PWB's Help system - even though there is a new Global Search option).

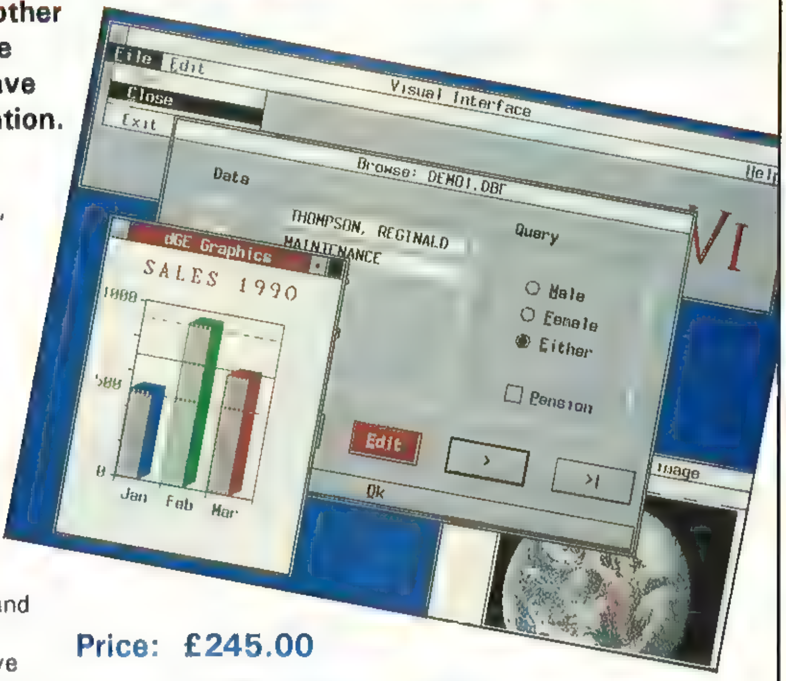
C++

The C++ compiler of C/C++7 is a full implementation of the AT&T version 2.1 language specification including support for true nested types (ie proper scoping of 'class within a class'-type declarations). Microsoft has also included prototype versions of more advanced C++ features such as templates and exception handling (Borland C++ V3.0 implements templates as part of the language, but has no mechanism for exception handling at all).

Exception handling is implemented as a system of macros which expand to C++ control structures and function calls. Although not identical to the proposed ANSI implementation, it should be relatively easy to port code written using the macros when the facility becomes part of the language proper. For example, in C/C++7:

VISUAL INTERFACE **GUI Library for Clipper**

We led the way with dGE. This is the other indispensable library for Clipper. For the first time your Clipper programs can have the look and feel of a Windows application.



Visual Interface has it all....

- Multiple windows that can be moved, sized, maximized and minimized
- A complete range of interactive controls
- An integrated event handler
- Simple to program
- Fully object-orientated
- Runs dGE graphics in it's windows

Your applications will never be the same....

- Create graphical dialog boxes, data-entry forms and context-sensitive help screens
- Select files from scrolling list boxes
- Import bit-mapped images
- Create queries using icons, radio buttons, and check boxes
- Run several windows concurrently and move between them with a keystroke or mouse-click

VI comes with a 250 page manual including over 100 pages of tutorial with examples.

Requires:
Clipper 5.01
DOS 3.x or higher
640K RAM
EGA or VGA

Price: £245.00

From: Bits Per Second Limited
14 Regent Hill, Brighton, E.Sussex BN1 3ED
Tel. 0273 727119 Fax 0273 731925

The VI library contains elements of Class(y) and SuperClass issued under licence. You do not need to buy these products to use VI with Clipper 5.01
Class(y) (c) Anton van Straaten
SuperClass (c) Chydale Software
dGE, Visual Interface (c) Bits Per Second Ltd.

CIRCLE NO. 460

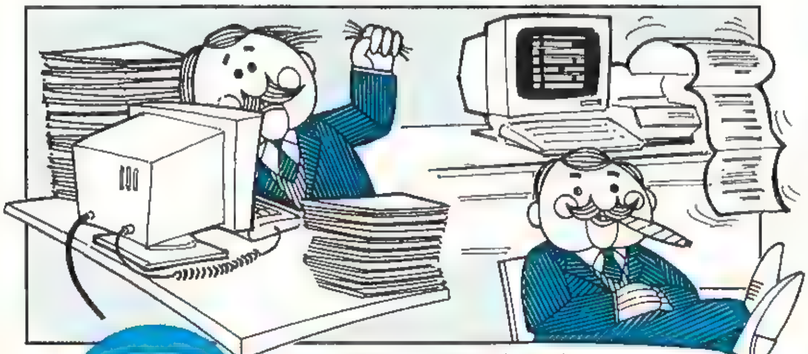
Put your PC on Autopilot . . .

AUTO RUN

PC Automation Software

Automate your PC to carry out all your repetitive tasks. For example, set Autorun to run Month End routines, Print reports, Sort databases, perform Mailmerges, Import and Export Data or Files across networks, Log onto Mainframes via modems, backing-up or processing data automatically at any time, day or night.
Autorun automation software will start-up, perform all assigned steps and terminate, no matter how complex, totally unattended.

- **Totally unattended operation**
- **Interactively learns the keys you press and the time between each keystroke**
- **Remembers Control, Alt and Shift keys**
- **Decision making capabilities within tasks**
- **Executes your tasks at any time, day or night**
- **Power failure recovery during automated processes**
- **Automatic Day and Date function**
- **Maintains time delay between keys**
- **Unlimited number of jobs and keys**
- **Fast and convenient editing of keystrokes**



PRICE
£99.95
[illegible]

Also available in Network version at
£299.95 Excl.

TURN YOUR PC INTO YOUR PERSONAL ROBOT!

WHAT THE PRESS SAY . . .

"Autorun is extremely easy for first time users to operate, the added features available to more experienced operators make it a very powerful suite of programmes indeed" — ***Micro Computer Mart**, April 1991*

"A lot of man hours can be freed up by this program . . . Autorun is a worthwhile program that will find a welcome home in many offices."
— ***Computer Shopper**, October 1990*

"Autorun adds up to a very powerful (and rather clever) time saving utility."
— ***What Micro**, April 1991*

UK DISTRIBUTORS:-
GUILDSOFT LTD.
GUILDSOFT DEPT.EXE 2THE COMPUTER COMPLEX,
CITY BUSINESS PARK, STOKE, PLYMOUTH, DEVON PL3 4BB
PHONE 0752-606200

GUILD Soft

CIRCLE NO. 461

ix

programming editor for Windows

◆ Full Windows implementation

Cut & Paste, nearly infinite Undo, Point and Click or key strokes control all features

◆ Multiple files and windows using MDI

Load and view several files simultaneously in separate windows

◆ Folding text

Manage large object modules with sections of text 'folded' up into a single line

◆ Code documentation

Add 'post-it' style notes to any file

◆ Fully reconfigurable keyboard and menus

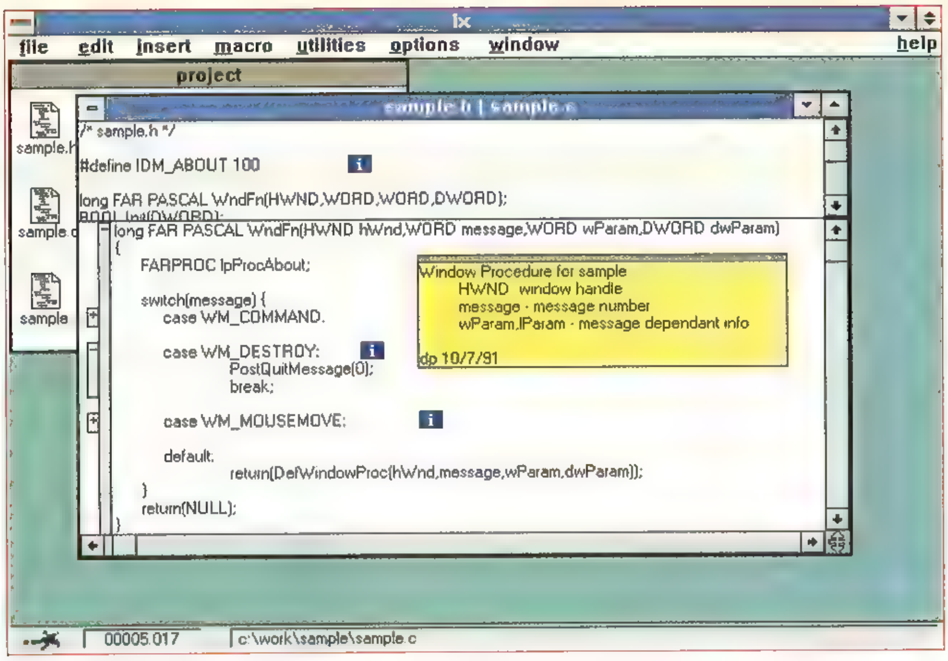
Re-assign all keys and menus, make multiple key assignments

◆ Powerful macro language built-in

Create and compile macros using any editor function with a Basic-like language

◆ £95 + VAT

positive



```
/* sample.h */

#define IDM_ABOUT 100

long FAR PASCAL WndFn(HWND,WORD,WORD,DWORD);
BOOL Init(DWORD);
```

```
long FAR PASCAL WndFn(HWND hWnd,WORD message,WORD wParam,DWORD dwParam)
{
    FARPROC lpProcAbout;

    switch(message) {
        case WM_COMMAND.

        case WM_DESTROY:
                PostQuitMessage(0);
                break;

        case WM_MOUSEMOVE:

        default:
                return(DefWindowProc(hWnd,message,wParam,dwParam));
    }
    return(NULL);
}
```

```
Window Procedure for sample
    HWND  window handle
    message - message number
    wParam,lParam - message dependant info

dp 10/7/91
```

As more and more users switch to Windows 3, the demand for Windows software continues to grow. Yet much of the developer's toolkit still runs under DOS. Writing for Windows has always been more complex than for DOS, and project management can be especially tricky as Windows programs tend to be made up of large numbers of files.

ix was developed to enhance programmer productivity. ix runs under Windows, and uses all the standard GUI features – MDI, split windows, cut & paste, choice of fonts and colours. It also contains several unique features which allow the developer far greater control.

With ix you can edit and test within Windows – no more switching in and out of a DOS editor. Compilation/make files can be launched from a 'Browse' window.

Browse windows help you keep track of the multiple files of a Windows project. A browser can be created for each project and used to fire up edit windows, launch resource editors (such as SDKpaint) and run programs as in Program Manager.

'Post-it' style notes can be stuck onto any source file – without modifying the file. Their names can be used to give a hypertext-like help system throughout your project.

And there's more; our AdHoc macro language, reassignable keys, custom menus and free telephone support, but we haven't got space to tell you about all that here. For more information, call or write us and we'll send you our information pack.

• positive limited •
• 22 westminster buildings •
• new york street •
• leeds ls2 7dt •
• 0532 343 104 •

Microsoft is a trademark and Windows is a trademark of the Microsoft Corporation. Positive Limited reserves the right to change the specification of ix without notice.

CIRCLE NO. 462

```
//*******************************************************************
//                    Borland C++ OWL Application
//*******************************************************************
#include <owl.h>
#include "winres.h"
extern "C"
{
  #include <stdio.h>
  #include <time.h>
  #include <string.h>
};
#define WM_MYMSG WM_USER + 1
//********************** Class Definitions **********************
class MyApp : public TApplication
{
public:
  MyApp( LPSTR AName, HANDLE hInstance, HANDLE hPrevInstance,
         LPSTR lpCmdLine, int nCmdShow )
  : TApplication( AName, hInstance, hPrevInstance,lpCmdLine,
                 nCmdShow ) {}
  void InitMainWindow();
};
class MainWnd : public TWindow
{
private:
  int     num_msgs;
  time_t  start;
  char    text[100];
public:
  MainWnd( PTWindowsObject AParent, LPSTR ATitle );
  void TimeMessages( RTMessage ) = [CM_FIRST + IDM_TIMECOUNT];
  void MyMsg( RTMessage )         = [WM_FIRST + WM_MYMSG];
  void WMPaint( RTMessage );
  void WMMouseMove( RTMessage )  = [WM_FIRST + WM_MOUSEMOVE];
  void WMActivate( RTMessage );
  void WMCreate( RTMessage );
  void WMHScroll( RTMessage );
  void WMLButtonDown( RTMessage );
  void WMMove( RTMessage );
  void WMSize( RTMessage );
  void WMVScroll( RTMessage );
  void WMDestroy(RTMessage);
};
//******************** Implementation code*****************
// Main program loop
int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR  lpCmdLine, int    nCmdShow )
{
  MyApp ma( "Borland OWL Application", hInstance, hPrevInstance,
            lpCmdLine,  nCmdShow );
  ma.Run();
  return ma.Status;
}
//======================
// Initialise main window
//======================
void MyApp :: InitMainWindow()
{
  MainWindow = new MainWnd( NULL, Name );
}
//==========
// MainWnd
//==========
//------------
// Constructor
//------------
MainWnd :: MainWnd( PTWindowsObject AParent, LPSTR ATitle )
: TWindow( AParent, ATitle )
{
  num_msgs = 0;
  strcpy( text, "Select menu item to start test" );
  AssignMenu( "MainMenu" );
}
//-----------------
// EVENT HANDLERS
//-----------------
void MainWnd :: TimeMessages( RTMessage )
{
  // initialise data
  num_msgs = 0;
  strcpy( text, "Processing messages..." );
  InvalidateRect( HWindow, NULL, TRUE );
  UpdateWindow( HWindow );
  time( &start );
  PostMessage( HWindow, WM_MYMSG, 0, 0 );
}
void MainWnd :: MyMsg( RTMessage )
{
  if ( num_msgs < 10000 )
  {
    num_msgs++;
    PostMessage( HWindow, WM_MYMSG, 0, 0 );
  }
  else
  {
    time_t end;
    time( &end );
    int elapsed = difftime( end, start );
    sprintf( text, "Time taken to process 10,000 messages = %d
         seconds",  elapsed );
```

Figure 4 - Borland OWL TIMER.CPP

```
//*******************************************************************
//                    Microsoft C++ MFC Application
//*******************************************************************
#include <afxwin.h>
#include "winres.h"
extern "C"
{
  #include <stdio.h>
  #include <string.h>
};
#define WM_MYMSG WM_USER + 1
#pragma hdrstop
//********************* Class Definitions **********************
class MyApp : public CWinApp
{
public:
  BOOL InitInstance();
};
class MainWnd : public CFrameWnd
{
private:
  int     num_msgs;
  CTime   start;
  CString text;
public:
  MainWnd();
  void afx_msg TimeMessages();
  LONG afx_msg MyMsg( UINT wParam, LONG lParam );
  void afx_msg OnPaint();
  void afx_msg OnMouseMove( UINT flags, CPoint point );
  void afx_msg OnActivate( WORD state, CWnd * pOther, BOOL bMinimised );
  int  afx_msg OnCreate( LPCREATESTRUCT pCreateStruct );
  void afx msg OnHScroll( WORD SBcode, WORD pos, CWnd * pScrollBar );
  void afx_msg OnLButtonDown( WORD flags, CPoint point );
  void afx_msg OnMove( int x, int y );
  void afx_msg OnSize( UINT type, int cx, int cy );
  void afx_msg OnVScroll( WORD SBcode, WORD pos, CWnd * pScrollBar );
  void afx_msg OnDestroy();
  DECLARE_MESSAGE_MAP()
};
//*********************** Implementation code
*****************************
// Instantiate global application object
MyApp theApp;
//======================
// Initialise main window
//======================
BOOL MyApp :: InitInstance()
{
  m_pMainWnd = new MainWnd();
  m_pMainWnd->ShowWindow( m_nCmdShow );
  m_pMainWnd->UpdateWindow();
  return TRUE;
}
//==========
// MainWnd
//==========
//--------------------------------------------
//       Create message map for MainWnd
//--------------------------------------------
BEGIN_MESSAGE_MAP( MainWnd, CFrameWnd )
  ON_COMMAND( IDM_TIMECOUNT, TimeMessages ) // menu message
  ON_MESSAGE( WM_MYMSG, MyMsg )              // user-defined message
  ON_WM_PAINT()                             // event handlers...
  ON_WM_MOUSEMOVE()
  ON_WM_ACTIVATE()
  ON_WM_CREATE()
  ON_WM_HSCROLL()
  ON_WM_LBUTTONDOWN()
  ON_WM_MOVE()
  ON_WM_SIZE()
  ON_WM_VSCROLL()
  ON_WM_DESTROY()
END_MESSAGE_MAP()
//------------
// Constructor
//------------
MainWnd :: MainWnd()
{
  num_msgs = 0;
  text     = "Select menu item to start test";
  Create( NULL, "Microsoft MFC Application",
          WS_OVERLAPPEDWINDOW, rectDefault, NULL, "MainMenu" );
}
//-----------------
// EVENT HANDLERS
//-----------------
void MainWnd :: TimeMessages()
{
  // initialise data
  num_msgs = 0;
  text     = "Processing messages...";
  InvalidateRect( NULL, TRUE );
  UpdateWindow();
  start = CTime::GetCurrentTime();
  PostMessage( WM_MYMSG, 0, 0 );
}
LONG MainWnd :: MyMsg( UINT, LONG )
{
  if ( num_msgs < 10000 )
  {
    num_msgs++;
    PostMessage( WM_MYMSG, 0, 0 );
  }
  else
  {
```

Figure 5 - Microsoft MFC TIMER.CPP

```
    InvalidateRect( HWindow, NULL, TRUE );
  }
}
void MainWnd :: WMPaint( RTMessage )
{
  RECT        rect;
  PAINTSTRUCT ps;
  HDC         hDC = BeginPaint( HWindow, &ps );
  GetClientRect( HWindow, &rect );
  SetTextAlign( hDC, TA_BASELINE | TA_CENTER );
  SetBkMode( hDC, TRANSPARENT );
  TextOut( hDC, ( rect.right / 2 ), ( rect.bottom / 2 ),
           text, strlen(text) );
  EndPaint( HWindow, &ps );
}
void MainWnd :: WMActivate( RTMessage msg )
{
  TWindow::WMActivate( msg );
}
void MainWnd :: WMMouseMove( RTMessage msg )
{
  TWindow::DefWndProc( msg );
}
void MainWnd :: WMCreate( RTMessage msg )
{
  TWindow::WMCreate( msg );
}
void MainWnd :: WMHScroll( RTMessage msg )
{
  TWindow::WMHScroll( msg );
}
void MainWnd :: WMLButtonDown( RTMessage msg )
{
  TWindow::WMLButtonDown( msg );
}
void MainWnd :: WMMove( RTMessage msg )
{
  TWindow::WMMove( msg );
}
void MainWnd :: WMSize( RTMessage msg )
{
  TWindow::WMSize( msg );
}
void MainWnd :: WMVScroll( RTMessage msg )
{
  TWindow::WMVScroll( msg );
}
void MainWnd :: WMDestroy( RTMessage msg )
{
  TWindow::WMDestroy( msg );
}
```

Figure 4 - Borland OWL TIMER.CPP (Continued)

```
    CTime     end     = CTime::GetCurrentTime();
    CTimeSpan elapsed = end - start;
    char buffer[100];
   sprintf( buffer, "Time taken to process 10,000 messages = %d seconds",
    elapsed.GetTotalSeconds() );
    text = buffer;
    InvalidateRect( NULL, TRUE );
  }
  return 0L;
}
void MainWnd :: OnPaint()
{
  CRect     rect;
  CPaintDC  dc( this );
  GetClientRect( rect );
  dc.SetTextAlign( TA_BASELINE | TA_CENTER );
  dc.SetBkMode( TRANSPARENT );
  dc.TextOut( (rect.right / 2), (rect.bottom / 2),
              text, text.GetLength() );
}
void MainWnd :: OnMouseMove( UINT flags, CPoint point )
{
  CFrameWnd::OnMouseMove( flags, point );
}
void MainWnd :: OnActivate( WORD state, CWnd * pOther, BOOL bMinimised )
{
  CFrameWnd::OnActivate( state, pOther, bMinimised );
}
int MainWnd :: OnCreate( LPCREATESTRUCT pCreateStruct )
{
  return CFrameWnd::OnCreate( pCreateStruct );
}
void MainWnd :: OnHScroll( WORD SBcode, WORD pos, CWnd * pScrollBar )
{
  CFrameWnd::OnHScroll( SBcode, pos, pScrollBar );
}
void MainWnd :: OnLButtonDown( WORD flags, CPoint point )
{
  CFrameWnd::OnLButtonDown( flags, point );
}
void MainWnd :: OnMove( int x, int y )
{
  CFrameWnd::OnMove( x, y );
}
void MainWnd :: OnSize( UINT type, int cx, int cy )
{
  CFrameWnd::OnSize( type, cx, cy );
}
void MainWnd :: OnVScroll( WORD SBcode, WORD pos, CWnd * pScrollBar )
{
  CFrameWnd::OnVScroll( SBcode, pos, pScrollBar );
}
void MainWnd :: OnDestroy()
{
  CFrameWnd::OnDestroy();
}
```

Figure 5 - Microsoft MFC TIMER.CPP (Continued)

```
TRY
{
  do_something();
}
CATCH( ExceptionClass, e )
{
  // e is an instance of the
  // exception object
  ...
}
AND_CATCH( ExceptionClass, e )
{
  ...
}
END_CATCH
```

System proposed by Stroustrup:

```
try
{
  do_something();
}
catch( ExceptionClass e )
{
  ...
}
catch( ExceptionClass e )
{
  ...
}
```

Template definition is also at a prototype stage and declarations must be converted to v2.1-compliant header files and .CPP files using a TEMPLDEF utility (for which the source code is provided). This is not too much of a hindrance, given that a number of common templates are provided in the class libraries shipped with the product, and programmers will not often need to define their own. As with exception handling, Microsoft plans to incorporate templates (and any other syntax) into the language proper only when an ANSI standard is available.

Class libraries

The availability of high-quality class libraries is perhaps the most important factor in taking full advantage of OOP and C++. With C/C++7, Microsoft has included an extensive range of DOS and Windows classes collectively referred to as Microsoft Foundation Classes (MFC - Figure 1). This group of classes is part of a wider hierarchy called Application Framework Extensions (AFX). The framework is incomplete at the moment (consisting only of the MFC group) but will be filled out in the compiler's next release. Full source code is included for all classes.

It can be seen that the MFC hierarchy is 'flat' or 'granular', meaning that there are not many levels in the tree. There are relatively few classes, but they have many methods. Most objects are derived from a superclass `CObject`. Unlike Borland's ObjectWindows Library (OWL), the MFC Windows classes encapsulate almost the whole of the Windows API, so one should not need to 'kickdown'. Microsoft has also provided an excellent range of general utility classes such as `CString` and `CTime`.

Dealing efficiently with message handling in GUI environments has become a moot point in C++ circles. The problem is that because Windows itself is not truly object-oriented you end up with enormous virtual function tables and are still left with a slow `switch` structure in a main window procedure (or 'dispatcher'). Borland got round the problem by a language extension whereby a Windows message number is associated with a virtual function in the

SOFTWARE PROTECTION!

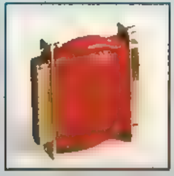
Can You Spot The Difference?

- ☑ High-level security keys
- ☑ Assembler-based, customer specific, encrypted interrogation routines
- ☑ Over 140 language interfaces available
- ☑ Compatibility - due to 9 years experience & 750,000+ keys sold
- ☑ Reliable on-going support
- ☑ Physically unique keys for each customer
- ☑ MS-DOS, OS/2, UNIX, XENIX, WINDOWS 2 & 3, "MACINTOSH"
- ☑ Transparent operation with most peripherals on market
- ☑ Free language updates
- ☑ Parallel, Serial, Mac ports

OVER 4,000 CUSTOMERS WORLDWIDE HAVE!

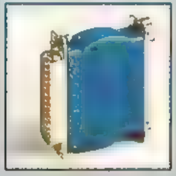
Electronic Key

The ideal device for identically produced software packages. Uniquely wired with customer code and a software code. Uses Assembler based program, decryption interface and random values.



1 Word Memory Key

Custom hardware wiring allows the developer total control over information stored in the key. 2 bytes of memory allows several packages to be protected with just the one key.



31 Word Memory Key

For multiple software protection schemes. 31 words of 16 bits of non-volatile dynamically programmable memory. Its capacity to store information provides virtually limitless power. Flexible protection scheme can be modified on-site during operation of software package.

UP3 Memory Key

Provides the ultimate in software security. Not tied to any language or O/S. 8 bit microprocessor powers from RS-232 level. Requires no power supply. For PC terminals, minis, & others using RS 232 C comms. Used on workstations. This key is effectively a computer.

Macintosh Key

Extremely powerful & customised protection for the Mac. 31 words available for random storage. Providing unequalled protection the Macintosh Memory Key connects to the SCSI 25 pin port and operates transparently.

MICROPHAR The European Leader*

* Based on the number of keys in use throughout Europe

Call us for more information or a demonstration package on

091-378 9191

CLEARSOFT,
Langley Moor, Durham DH7 8HG.
Telephone: (091) 378 9191 Fax: (091) 378 9393

Other European Countries

- **FRANCE**, Tel: 33-1-47-38-21-21
- **BELGIUM**, E2S (091 21 11 17)
- **GERMANY, AUSTRIA:** (06223 - 73730)
- **HUNGARY**, Polyware Kit (76-22-307)
- **ITALY**, Siosistemi (030 24 21 074)
- **PORTUGAL**, HCR (1 56 18 65)
- **SPAIN**, Hal 2000 (032 37 31 05)
- **SWITZERLAND**, SAFE (024 21 53 86)
- **THE NETHERLANDS**, E2S (015 15 88 37)

CIRCLE NO. 463

class declaration (*see .EXE Nov '91*). Microsoft has taken a slightly different approach by using a system of macros and the addition of the `afx_msg` keyword. The keyword acts a bit like the `virtual` method modifier but does not have its implicit behaviour when it comes to inheritance - the macros you use have to tell the compiler what is the object's immediate base class.

Using Borland's OWL library, the array of pointers to a window's event handlers and their associated message numbers is generated by the compiler and held as a Dynamic Dispatch Virtual Table (DDVT) adjacent in memory to the class's ordinary Virtual Method Table (VMT). With MFC, the `DECLARE_MESSAGE_MAP()` macro declares a similar array of pointers and numbers within the class declaration. This array is then subsequently initialised in a `BEGIN_MESSAGE_MAP()` macro block. When a window receives a message, a small assembler routine searches the array and matches the number with a function pointer in a similar way to Borland's OWL.

OWL vs MFC

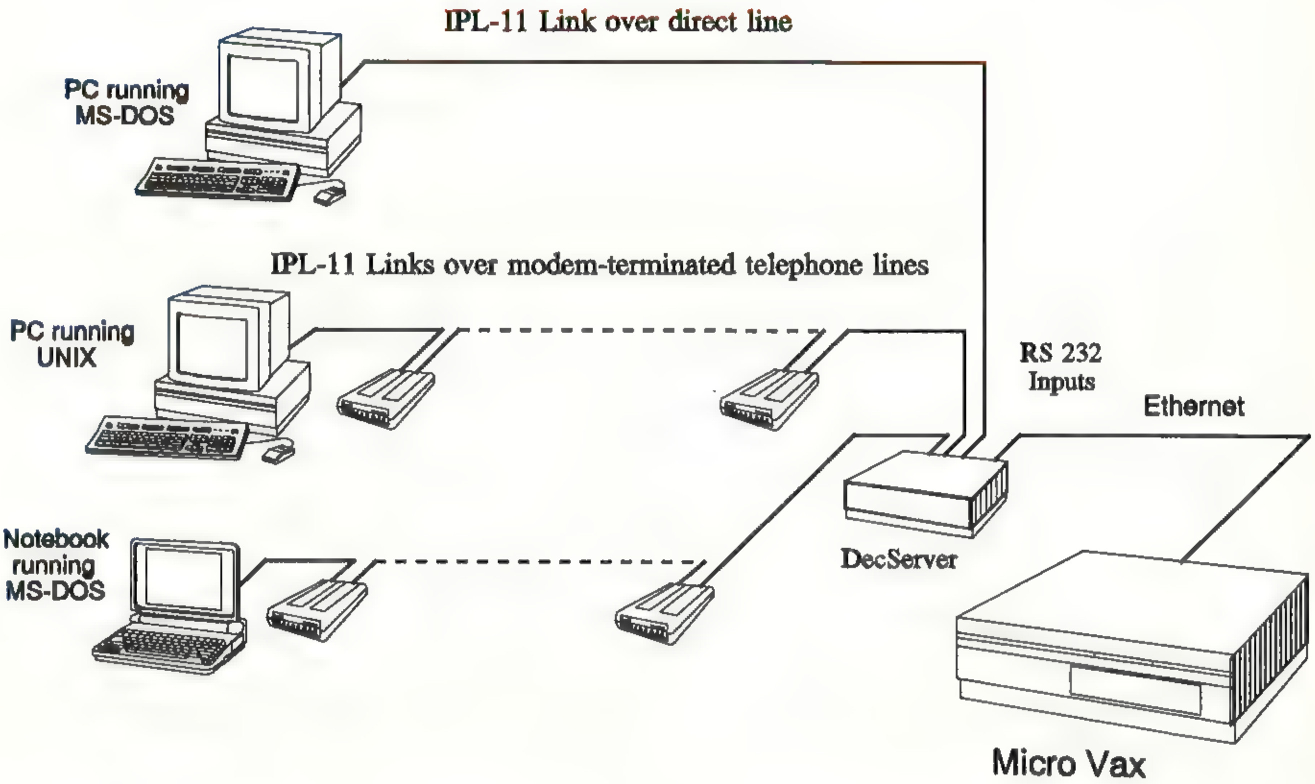
In order to test the relative efficiency of the two message routing systems and to examine the ease of Windows development using the class libraries, I devised a simple application. The program is called TIMER and it measures the time taken (in seconds) to process 10,000 user-defined messages. The timer is kicked off by selecting a menu item, and the elapsed time is displayed in the centre of the client window area. A number of simple information messages are also displayed. The source code for OWL and MFC versions of this application are listed in Figures 4 and 5. To make the test a bit more realistic I inserted a number of extra 'dummy' event handlers to generate a reasonably sized message array. I also tested the programs 'under stress' by repeating the timer experiment while moving the mouse pointer rapidly over the client window, thus generating thousands of `WM_MOUSEMOVE` messages.

The results of this test are tabled in Figure 6. It can be seen that the MFC application did not appear to process the messages as rapidly as its OWL equivalent. I am wary of drawing too many conclusions from this because it is such a simple application, but it may indicate an area where Microsoft could make improvements. Microsoft compile and link times were also slower but .EXE sizes were considerably smaller than their OWL counterparts. This is mainly due to Microsoft's new 'smart-method' linker (similar to that supplied by JPI) which strips out unused methods from the executable. The MFC source was slightly longer, primarily because of the need to declare the message map entries. This I found a bit annoying because it means that the programmer effectively has to declare a window's event handlers twice - once in the class definition and again to initialise the message map. Message handling functions in MFC also have distinct prototypes (ie they receive different parameters and return different values). This is good in one respect because it means that the standard Windows message parameters have already been decoded into meaningful data (eg a mouse coördinate). A downside of this is that it is necessary to look up and declare the exact prototype of an event handler you wish to override.

Other matters

The PWB has a very powerful class browser. Browse information is generated at compile time by a set of compiler options. These cause the creation of separate .SBR files which are then 'compiled' to a .BSC browse database with the BSCMAKE utility.

File Transfer Software to link PC's, UNIX, VAXes and PDP-11s



The XOREN IPL-11 range of file transfer software packages allows simultaneous error-free file transfers between multiple PCs running DOS or UNIX and any VAX, PDP-11 or UNIX system. No special communications hardware is required. Each PC's comms port (com1, com2 etc.) may be connected directly or via a telephone line to a standard RS232 port on either a terminal server or the host computer

Since IPL-11 operates on a peer to peer basis, any PC, VAX, PDP-11 or UNIX system installed with IPL-11 can exchange files with any other PC, VAX, PDP-11 or UNIX system installed with IPL-11.

XOREN

Xoren Computing Ltd
28 Maddox Street
London W1R 9PF
Telephone 071-629 5932
Fax 071-629 5432

CIRCLE NO. 465

Graphical User Interface CLEARING HOUSE

"Europe's Largest Distributor" for Microsoft Windows and OS/2 graphical environments

Thousands of Windows and OS/2 Applications to Choose From...

File Edit Search Order!

(44) 03552-64888

Order by FAX (44) 03552-64777

Monday-Friday 9:00am to 8:00pm

GUI Clearing House is the ONLY EUROPEAN reseller specializing exclusively** in Windows and OS/2 Presentation Manager environments. This specialization is necessary in order to properly serve you, our customer, in this rapidly evolving market. **Our goal is to provide you with the finest customer service and technical support in the industry and to ensure that the products you select are the correct ones for your application. CALL TODAY!

WINDOWS APPLICATIONS

General/Utilities

Adobe Type Manager	$69
Asymetrix Toolbook	$299
Becker 2.0+	$99
CO/Sessions/Windows	$139
Distinct Back-up	$119
File F/X	$49
Icon Pak II	$59
Intermission	$39
More Windows	$89
PackRat	$279
Personal Librarian	$659
PLUS for Windows	$179
Publisher's Powerpak	$69
PubTech File Organiser	$89
Sitback Windows	$119
SmarText (Lotus)	$409
Whiskers Pro	$25
WideAngle/Windows	$89
Windows Shopper's Guide	$16

Publishing/Graphics

Adonis Clip-Art Window Shop	$42
Ami' Professional 2.0 upgrade	$99
Ami' Professional 2.0	$309
Ami'Pro & 123 Windows upgrade	$169
Archetype DESIGNER	$795
Arts & Letters Graphics Editor	$509
CA Cricket Graph v1.3	$109
Corel DRAW! 2.0	$389
DoDOT! (capture/convert)	$119
Gram-mat-ik for Windows	$69
HiJaak (Inset Systems	$89
Image-In-Colour	$649
Image-In-Colour Full Pack	$749
Image-In-Colour Upgrade	$249
Image Prep 4.0	$239
Import for Windows	$259
Instant ORGCharting	$139
Language Master for Windows	$89
Micrografx Charisma	$349
Microsoft Office for Windows	$499
Picture Publisher	$469
PowerPoint for Windows	$319
Publisher's PaintBrush	$329
Ventura Gold Series	$549
VentanaDraw	$135
WinDraw (Addison Research)	$89
WordScan Plus	$699
Word for Windows	$319

Financial/Accounting

Cortex Business Manager	£299
Braincel	$229
Computerized Classic Accounting	call
Crystal Accounting	$209
Excel for Windows (Microsoft)	$319
Lotus 123 for Windows	$439
Wingz (Informix)	$359

Connectivity

DaVinci eMail for Windows	call
DynaComm (Sync or Async)	$259
DynaComm Elite	$399
EXTRA! for Windows	$379
Faxit! V2.0 for Windows	$129
Microphone II	$259
Net Magic (US Version)	$589
Network Courier 150 user	$509
ReachOut (modem version)	$149
Remotely Possible 6 user	$249
ShareFAX	$799
Terminal Plus (FutureSoft)	$79
TERM for Windows	$179
Windows Workstation	$559
WinComm	$139
WinTerm	$409

Member

WPMA

The Windows & Presentation Manager Association

Database/Forms

Access SQL (SPI)	$1279
dBFast/Windows	$299
ERwin ver 1.5	$849
INTEGRA SQL	$609
JetForm	$399
MARVIN Image Database	$2395
PerForm Pro	$299
Q+E ver 2.6, single user	$179
Superbase 2 ver 1.3	$249
Superbase 4 ver 1.3	$439
Superbase 4 Developers Ed	$899
Chart Builder for Superbase	$169
Thinx	$399

Engineering/Planning

ABC Flowcharter	$199
Design/IDEF	$2795
DesignView	$749
DRAFIX Windows CAD	$459
MathType	$199
MetaDesign	$329
Milestones, Etc	$169
Project for Windows	$449
RFFlow, flowcharting	$119

Development Tools

ACTOR 4.0 Professional	$385
ADEPT Expert Appl Builder	$595
ICEWorks Pro (Icon Editor)	$79
Borland C++	$329
Borland ObjectVision	$99
Borland Turbo Pascal	$199
Bridge 2.0 Toolkit	$569
C-TRIEVE/Windows	$349
C++/Views	$469
Bluesky Windows Maker	$995
WinImage Image Toolkit	$279
Icon Pak Developers Edition	$395
Knowledge Pro Windows	$429
Lasertalk PC	$309
Microsoft SDK for Windows 3.0	$319
Multiscope for Windows 3.0	$259
ObjectGraphics (WhiteWater)	call
ProtoView	$629
ProtoGen	$189
Pubtech BatchWorks	$69
SmallTalk V/Windows	$319
SpeedEdit for Windows	$229
Whitewater Resource Toolkit	$119
WindowsMAKER Pro	$775
WinTrieve	$299
Zortech C++ Windows/DOS	$369
Zortech C++ Dev Edition 3.0	$499

Dynamic Link Libraries

Graphics Server SDK	$479
Halo Image File Format Lib	New
PowerLibW (ETN)	$229
PowerShoW (ETN)	$395
Q+E Database Library	$309
Toolbox (Drover)	$249

FAST Movers

Access to Platinum (special)	$149	Cortex Business Manager	£299
PictureBox F/X	$279	Microsoft Word 2.0 for Windows	New
Braincel	$229	Microsoft Works for Windows	$149
Carbon Copy Plus 6.0	$69	Norton Desktop	$99
Clip N Save Windows or OS/2	$129	Pubtech File Organiser	$89
File Shuttle XPRESS for Laptops	$79	Realizer	$299
ICEWorks Professional	$79	Scrapbook+	$119
Image-In-Colour (Full Pak/Upgrade)	call	Snap It (special)	$49
Just Write	$139	Superstore (AddStor)	$$119
Liana (object oriented lang.)	$495	Terminal Plus (FutureSoft)	$79
Micrografx Windows DRAW	$119	WordPerfect for Windows	New
Microsoft Money for Windows	$55	Visual BASIC	$159

OS/2 PM APPLICATIONS

Access to Platinum for OS/2	$199
Arts & Letters for PM	$339
Asymetrix ToolBook for OS/2	$229
Brief for OS/2	$229
ChipChat (Cawthon)	$279
Clip3D Library for PM	$89
Collage PM	$159
CorelDraw! 2.0 for OS/2	$399
DeScribe	$399
Excel for OS/2	$499
HyperACCESS/5 for OS/2	$169
Lotus 123/G	$579
Nexpert Object, PM Runtime	$929
Object-1/PM (MDBS)	$895
OS/2 PM Softset (Microsoft)	$109
OS/2 PM Toolkit (Microsoft)	$329
Packrat for PM	call
Perspective for OS/2	$259
PM Assistant	$115
Print Screen for OS/2 PM	$72
Q+E/PM ver 2.6 Std Edition	call
Q+E/PM ver 2.6 Server Edition	call
Relish	$139
Sidekick for PM	$169
SmallTalk/V for PM	$319
SpeedEdit	$255
Super Project Expert for OS/2	$599
Systos Plus/PM	$319
Ventura Gold for OS/2	$529
VZ Programmer/PM	$1225
WideAngle for PM	$89
Wingz	$359
Word for OS/2	$319
WordPerfect 5.0 for OS/2	$339
WinPRO/PM (Xian)	$379
Zortech C++ Dev Edition 3.0	$499

Featured Products

PictureBox For Windows
(by Highland Grafix)

PictureBox is a MultiMedia Authoring and Presentation application with the scope to meet your most demanding needs. At a price accessible to all, PictureBox is designed to make developing in-house Multi-Media presentations easy and quick. Supports live video, user DLL's, Metafiles, stills, sound, graphics and text. No programming required.

Cost to you £99.00

WinImage Toolkit
(by Highland Grafix)

WinImage is an easy to use, customizable, developers imaging DLL. Complete with full documentation and is Royalty Free! Supports all Windows compilers.
Image support includes
TIFF,PCX,EPS,Bitmaps,GIF,Metafiles
numerous special effects F/X,
animation and much more.

Cost to you £159.00

Prices to you are calculated on US Dollar price, above, +15% +UK Shipping + VAT

PAYMENT-Visa, Mastercard and American Express cards accepted. Personal and company cheques accepted. Please allow 2 weeks for processing. Corporate and institutional purchase orders subject to credit approval and a minimum initial purchase of £500, paid via COD or in advance. COD minimum purchase is £100, maximum £1,000 payable by cash, cashier's cheque, or money order. SHIPPING-Shipping/handling charge is 3% or £6.00 minimum;£5.00 extra for COD. Larger shipments may require additional charges. All shipments FOB East Kilbride,UK. All duties are the responsibility of the purchaser. Call for information for shipments outside UK.

TERMS-All products carry manufacturer's warranty only. Product warranties, guarantees, rebates, trial perid priveleges, or other promotional programs are handled by the manufacturer. Defective items replaced or repaired at our discretion. All items returned must be accompanied by a return material authorisation number (RMA). There is a 15% restocking fee on non-defective merchandise. Shipping is not refundable. Opened software and other consumables are non-refundable. We cannot guarantee compatibility. Item availability, price, terms, and policies subject to change without notice. GUI-Euro Ltd. is not liable for damage due to omissions or typographical errors. Mailing address is GUI-Euro Ltd, 18, Albion Way, East Kilbride G75 0YN. Copyright 1990,1992 GUI-Euro Ltd. all rights reserved.

CIRCLE NO. 466

Because the browse information is not contained in the .EXE file, there is no overhead and a great deal of detailed cross-reference information can be stored. In Turbo C++ for Windows, similar data is burned into the executable and is not nearly as detailed as that available in C/C++7. With the PWB's class browser you can: display a list of all classes and their members in a 'tree' format (this is useful but not as good as a true graphical view); view all classes sorted by protection and scope; view all relationships between classes, class symbols, objects and member functions. Figures 2 and 3 show the PWB's class browser in action.

Conclusion

Microsoft has been very careful not to jump the gun in the race to C++ - one might even suggest that they have been slow off the blocks. But C++ is still an evolving language with many features not yet standardised. Microsoft's goal has been to produce a robust, reliable and above all *correct* implementation of C++, which is a highly complex language with many pitfalls for compiler-writers.

First impressions indicate that, despite improvements, C/C++7 is still a bit on the slow side with regard compilation speed, but I must state again that I was dealing with a beta version. I was also expecting great improvements in the PWB but was somewhat disappointed.

Compiler C++ Windows Library		Turbo C++ for Windows ObjectWindows (OWL)	Microsoft C/C++ 7.0 MS Foundation Classes (MFC)
Compile & link time	-pch -debug	55s	1m 15s
	-pch +debug	1m 11s	2m 20s
	+pch -debug	40s	47s
	+pch +debug	55s	1m 32s
.EXE size (bytes)	-pch -debug	57344	34304
	-pch +debug	127254	67584
	+pch -debug	57344	34304
	+pch +debug	127063	67584
Number of lines of code		162	168
Time taken to process 10,000 messages (secs)	-mouse mvt	8	10
	+mouse mvt	10	13

NOTE: pch = pre-compiled header
Data for a 25 MHz / 386-based PC with 4 MB RAM

Figure 6 - Comparison of OWL and MFC versions of TIMER

The class libraries supplied are very extensive and contain heaps of code to aid debugging. It does, however, seem a shame that vendors (Microsoft and Borland) have had to resort to language extensions/macros to implement compiler-dependent solutions to the 'Windows Problem'.

Worth the wait? Yes, I think so. Microsoft's commitment to producing code of the highest quality should ensure that C++ becomes more widely used for major development projects. As always, there is room for improvement, but I think the release of C/C++7 will herald the true dawn of C++.

EXE

Microsoft C/C++ version 7.0 will cost £335. Existing users of C 6.0 or (wait for it) any other C/C++ compiler (eg Borland, Zortech, JPI etc) can get a copy for £150. C/C++7 should start shipping at the end of February. Microsoft is on 0734 500741.

Programmer's Editors

They say that a programmer's editor is his best friend. Cliff Saran looks at five offerings.

A Programmer's Editor is probably the least considered programming aid. Before the birth of the PC, mainframe programmers had to rely on a systems editor. Today, the evolution of the standalone editor has produced a highly configurable beast, offering an environment for editing and compiling that is superior to its IDE cousins. In this article, I shall be looking at five Programmer's Editors varying considerably in price, complexity and the features that they have to offer.

QEdit

While I was compiling a list of editors for this article, I came across a rather good-looking shareware offering called QEdit. QEdit provides many features that put it above more basic editors.

First, it is fast. Even the largest file it could handle (about 490 KB) in my DOS setup it took milliseconds to load. Figure 1 shows a benchmark test that I ran on all the editors in this article. Buffer size is limited by the amount of free memory on your machine. In fact, you can open as many files as you like, as long as you have enough free memory left. Since this memory pool is central to the functionality of QEdit, if you are using a large Scratch buffer (eg during a Cut/Copy and Paste operation) it is best to flush the buffer once you've finished using it (by copying something smaller into it).

... Epsilon is the only editor to support concurrent processing

	QEdit	Ed	Epsilon	Brief	PVCS
Manufacturer	SemWare	Soft as it Gets	Lugaru's Software	Solution Systems	Polytron/ Sage
Largest Buffer Size	497 KB	100 MB	x	x	x
Largest Cut/Copy Block	270 KB	x	x	x	x
Resident in DOS shell	60 KB	8 KB	5 KB	6 KB	6 KB
Maximum Line Length (chars)	512	1024	x	512	x
Maximum Number of Lines	15868	65000	x	x	x
Maximum Number of Windows	8	8	256	32	256
Maximum Number of Open Files	-	30	x	x	x
Save Session info	No	Yes	No	Yes	Yes
Cut/Copy Block delay (secs)	x	5	3	4	x
Paste Block delay (secs)	x	75	x	37	x
Search & Replace delay (secs)	7	155	25	18	28
WordStar Configuration	No	Yes	No	No	Yes
Brief Configuration	No	Yes	No	Yes	Yes
Menu System	Yes	Yes	No	No	Yes
Keyboard Macro	Yes	Yes	Yes	Yes	Yes
Macro Language	No	Yes	Yes	Yes	Yes

- - Does not apply
x -Insignificant

The above benchmark was performed using a 40MHz 386 with 4Mb of RAM running MS-DOS V5.0. The **Cut** and **Paste** times were found by timing how long the editors took to perform these operations using a block size of 70KB. The **Search and Replace** times indicate the time taken for each editor to perform 4096 replacements (replacing the string 'ALongWindowsVar' with 'AShortWindowsVar'). The values for the 'Resident in DOS Shell' show how much memory the editor uses when running in a DOS shell.

Figure 1 : Programmer's Editors Benchmark

Second, it is easy to use. There's a menu system (activated with <ESC>), a status bar and a handy pop-up help screen that shows the key sequences for those frequently used operations (like Loading files or Cutting and Pasting). QEdit provides three ways to mark a block of text. A block may be marked as a group of rows or columns. Moving columns provides the easiest way to align a group of comments with the right margin or to add another level of indentation to a block of code in a loop construct. The block may then be manipulated using a single key stroke (eg the grey '-', '+' and '*' keys perform cut, copy and paste operations respectively).

Since QEdit is quite a small program, using only 58 KB of memory, it leaves enough room in a DOS shell to run your favourite compiler. There is no way to trap compilation errors but the ability to compile within an editing session is a great advancement over using a command-line compiler - at least when you return to your editing session it will remain in the same state as you left it (this is essential in QEdit as it doesn't save session info when you quit).

QEdit is configurable. It uses a separate program (QCONFIG.EXE) to change the working environment, and it has a keyboard macro recorder. It supports C-style indentation and there's also a built-in parenthesis/bracket matching macro. Unfortunately, it is not possible to change the printer port - printer output is *hard-wired* to `LPT1`.

QEdit provides all of the basic requirements of a good Programmer's Editor. There's limited support for C programming style and compiling within the editor (although it doesn't highlight the erroneous line like some of the more up-market offerings).

START THE YEAR WITH THE LATEST TOOLS AT THE LOWEST PRICES

PRE-SALE CONSULTING

POST SALE SUPPORT

NEW & NOTEWORTHY

Baler v5.1 Converts worksheet files into standalone, executable programs. Drastically reduced price. Order while supplies last!	£115
Borland C++ v3.0	£206
Dr DOS v6.0	£55
Drovers Professional Toolbox v2.0 for Windows	£180
Laplink Pro v4.0	£74
Microsoft OS/2 Presentation Manager Toolkit v1.21 Special reduced price! Order while stocks last!	£250
Microsoft Publisher for Windows	£106
DeskTop Publishing for Windows	
Microsoft Quick C for Windows	£93
Microsoft Windows v3.1	CALL
MS CD-ROM Programmer's Library Contains over 70 indexed and cross referenced books and reference manuals, with over 20MB of code that can be included in your programs.	£214
MS Visual BASIC Create GUIs in minutes without writing a line of code! Then add functionality with a powerful general purpose language!	£94
MS Word v2.0 for Windows	£299
NE-2000 Ethernet Kit 16 Bit high performance network adapter.	£135
Netware Lite	£69
Norton Desktop for Windows Now includes PL/SQL and SQL*DBA.	£75
QDOSIII Edit up to five, 1MB files simultaneously. Mouse support. Windows compliance, NetBios Compatible	£38
Spontaneous Assembly Drastically reduced price. Order while supplies last!	£85
Turbo Pascal for Windows Special price 'til Jan 31st. Get your copy now!	£133
ViVa 7496if 9600bps FAX/Modem 9600bps internal Send/receive fax with modem	£140
ViVa 2400 Internal Modem	£47
ViVa 2400 Internal Modem, MNP 5	£76
Zortech C++ Developer's Edition v3.0 Optimising C++ Compilers for DOS, Windows & OS/2	£339
Zortech C++ v3.0 for DOS, Windows and OS/2	£195

ARTIFICIAL INTELLIGENCE

Arity Combination for Windows	£736
Arity/Prolog Compiler & Interpreter v6.0	£398
Arity Windows Toolkit	£191
PDC Prolog (Turbo Prolog)	£177

ASSEMBLERS & LINKERS

Blinker	£177
Dis/Doc Professional	£166
Microsoft Macro Assembler v6.0	£62
Sourcer 486 v4.0	£109
Spontaneous Assembley v2	£99

BASIC

Mach 2 for MS QuickBASIC	£48
Object Script QuickBASIC	£93
PowerBASIC	£62

BUSINESS

ABC Flowcharter	£156
Adobe Illustrator for Windows	£239
Baler v5.1	£115
Form Tool Gold	£48
Grammatik IV	£47
IBM Current	£187
IBM Storyboard Live	£243
Macrografx Charisma	£280
ORG PLUS Advanced	£53
Quattro ProCompetitive Upgrade	£78
Show Partner F/X	£198
SideKick Plus	£110
Super Project Plus	£210
Xeron FormBase/Windows	£262

C

Aztec C Cross Development Systems	£331
Borland C++ v3	£206
C Communications Toolkit	£114
C-terp 386 with Source	£169
CLEAR+ for C	£106
Control Pak/Windows	£338
DESQview API C Library	£97
Essential Communications	£171
Greenleaf Comm++	£132
Greenleaf ViewComm	£370
Informix C-ISAM for DOS	£130
JPI Topspeed C++ ProEdition	£225
Lattice C Development Sys v6.0	£221
Microsoft C PDS v6.0	£221
Object Script C	£308
PC-Lint	£66
Pro-C with Workbench	£491
Turbo C TOOLS	£84
Turbo C++ Turbo Vision101	
Watcom C/386 v8.5	£
WATCOM C 386 Professional	£
WindowsMaker	£525
XVT (Windows/PM/Macintosh)	£366
Zortech C++ Database Class Library	£170
Zortech C++ v3 Windows, DOS and OS/2	£195

CADD

3D Drafting	£175
AutoSketch	£126
Design CAD 2D	£128
Design CAD 3D	£170
DRAFIX CAD for Windows	£362
DRAFIX CAD ULTRA	£209
Generic CADD	£248

CASE

Case: PM for C++	£1,511
Case: Windows Corporate Edition	£754
Case. Windows Standard Edition	£375
Compeditor II	£199
EasyCASE Plus v3.0	£375
Oracle CASE Dictionary	£1,326

COBOL

COBOL sp11	£366
COBOL/2 Compiler with Toolset	£1,278
Dialog System	£710
Level II COBOL	£351
Microsoft COBOL	£364
RM/COBOL -85 Develpment System	£667

COMMUNICATIONS

Carbon Copy v6.0	£96
Crosstalk for Windows 3.0	£103
LapLink Pro v4.0	£74
Mirror III	£71
Procomm Plus	£57
Reflection for Windows	£755
Remote 2 (Host & Caller)	£93
SmarTerm 240	£178
WinConnect	£51

DATABASE & FILE MANAGEMENT

Advanced Revelation	£622
ALPHA/Four competitive upgrade	£80
AskSam	£187
Btrieve DOS 3.1 Network for PC	£338
Btrieve for Windows	£338
Clarion Personal Developer	£38
Clarion Professional Developer	£392
DataFlex	£347
DataFlex Full Development System v3.0	£317
dbFast/PLUS	£147
dbFast/Windows	£211
DGE - Database Graphics v4.1	£177
FairCom Toolbox Special Edition	£395
Gupta SQL Windows	£803
Informix SQL for DOS	£459
Oracle Tools & Database v6.0	£849
R&R Clipper & FoxBASE+ Modula v3.0	£34
R&R Relational Report Writer for dBASE v4.0	£159
R&R Report Writer for Oracle SQL v1.0	£189
R&R Worksheet Report Writer v1.0	£95
R&R COde Generator v4.0	£129
R-BASE Single User	£414
SQL Windows for Btrieve	£1,208
Superbase 4 Windows	£368
UI Programmer	£282
XQL Relational Data Manager	£467
Xtrieve Plus	£355

DEBUGGERS

ClearView 386/486 Debugger	£280
Multiscope Debugger for DOS	£100
Multiscope Debugger for Windows	£212
Periscope II with Breakout Switch v5.0	£107
Turbo Debugger & Tools v2.0	£85

DESKTOP PUBLISHING

Adobe PostScript Cartridge	£262
Adobe Type Align/Windows	£52
Adobe Type Manager for Windows	£53
Catch Word OCR	£120
ColoRIX VGA Paint	£85
Facelift for Windows	£51
Fontware Installation Program	£49
Fontware Typeface Packages	£68
Go Script Plus	£158
Inset + Hijaak	£145
OmniPage Pro.	£527
OmniProof for DOS	£87
PageMaker	£435
Persuasion for Windows	£767
Softcraft Font Editor v1 15	£158
WordScan 286/386 2 Mb	£338
Xerox Ventura Publisher Gold v4.0	£CALL

DISK/DOS/BACKUP

Back-It	£65
Copy II PC	£23
Disk Manager	£31
Norton Commander	£83
Norton Utilities	£97
OI/Tune	£38
PC Tols with Anti Virus	£85
SpeedStor Universal	£54
SpinRite II	£51
WonderPlus	£42
XTree Pro	£47
XTree Pro Gold	£79

EDITORS

Vrief for C++	£169
Brief v3.1	£155
BDT+	£187
Epsilon Programmer's Editor for DOS,OS/2 or UNIX	£152
KEDIT	£85
Norton Editor	£42
Sage Professional.	£167
VEDIT PLUS	£97

FORTRAN

Fortran Toolkit 2	£102
GRAFMATIC & PLOTMATIC	£136
Microsoft Fortran PDS v5.1	£204
RM/FORTRAN Development System	£304
Spindsite Utility Library (F77L)	£70
ULI Maths Functions Library (F77L)	£166

GRAPHICS

CorelDraw	£329
Dodol	£98
Draw Applause	£232
GRASP with ARTOOLS	£169
GSS GKS Kernal System for DOS	£366
GSS Graphic Devlpm Toolkit for DOS	£489
HALO	£200
HALO Professional	£301
HALO Windows Toolkit	£298
Harvard Graphics	£318
Hijaak v2.1	£109
MetaWINDOW	£96
PC PaintBrush for Windows	£45
PCX Programmers Toolkit	£700
Pizazz Plus	£75
Windows Draw Plus	£141

HARDWARE

ASCII Internal 2400 Modem	£60
ASCII Internal 9600bps v32 with MNP5	£378
FastMath 83887-16 Math coprocessor	£214
Logitech TrackMan Serial Mouse	£67
Paradise 1024x768 VGA w/512k	£169
Practical 2400 Pocket Modem	£94
ProScan PS1000 64 Greyscale Scanner	£458
SatisFAXtion Classic	£330
ScanMan Plus PC	£151
The Complete Communicator	£364
Worldport 2496 Portable Fax	£410

INTEGRATED

Framework IV	£388
Lotus Symphony	£428
Lotus Works	£79
Microsoft PC Works	£81

LAN

LANtatstic AE-2 Ethernet Starter Kit	£397
NE-100 Ethernet Kit	£110
Netware v3.11 20 user	£2,052
Netware C Interface for DOS	£175
Oracle for Server	£1,106
Procomm Plus for Network	£287
Q-DOS II for Network (8-user)	£40
Saber File Manager	£176
Saber LAN Administration Pack	£310
Saber Menu for MS Windows 3.0	£220
Saber Menu System Sungle-User	£40
Saber Meter Usage Tracking Software	£87
Saber Secure	£66
SoftwareBridge LAN	£126
Super Project Expert Lanpak	£370
Superbase 4 Windows Network	£527

MAC

Adobe Illustrator Mac	£293
Adobe Type Manager	£49
Caron Copy Mac	£96
LapLink Mac III	£74
MathCad	£269
Mathematica	£379
Norton Utilities for Mac	£72
Oracle Developers Kit Mac	£442
Pagemaker.	£429
PC-Matlab/Mac	£663
Persuasion	£275
Quark Xpress	£452
Smalltalk/V for Mac	£95
Zortech C++ for Mac	£173

MEMORY MANAGEMENT

386 Max	£53
Headroom	£62
QEMM 386	£50
Turbo EMS	£43
Vcache	£40

MODULA-2

EmsStorage with Full Source	£43
Logitech Modula-2 Development System	£139
Logitech Modula-2 Window Pack	£27
ModBase	£19

OBJECT ORIENTED PROGRAMMING

C++ Views	£281
Codebase++	£196
HCR/C++	£641
Smalltalk/V v3.0	£61
Smalltalk/V Windows, Technomatic & CBP	£
Smalltalk/V 286	£95
Smalltalk/V EGA/VGA Colour Pack	£24
Smalltalk/V-Windows	£237
Zinc Interface Library	£141
Zortech C++ MS-DOS 386	£565
Zortech Windows Toolkit	£40

OPERATING ENVIRONMENTS

386/ix Operating System Release 3.2	£293
DESQview	£68
GeoWorks Ensemble	£93
OS/2 Extended Operating System	£584
OS/2 Standard Operating System	£227
SCO UNIX System V/386 v3.2.2	£377

OS/2

Corel Draw for OS/2	£282
Describe Word Publisher (Client)	£145
Display Write 5/2	£206
Halo for OS/2	£298
Microsoft OS/2 Presentation Mgr Softset	£81
Microsoft OS/2 Presentation Mgr Toolkit v1.21	£199
Oracle OS/2	£575
PageMaker for OS/2	£575
Paradox OS/2	£407
Qedit for OS/2	£43
QEMM Expanded Memory Manager 50/60	£51
Smalltalk/V-PM	£237
Super Project Expert/2	£424
ToolBook (OS/2)	£247
TopSpeed C++ for OS/2 Standard	£187
Vitamin C for OS/2	£281
Wide Angle for OS/2	£55

OTHER

ACT (Sales lead tracking)	£205
Adlib Card (sound)	£92
Builder (Batch Compiler)	£71
Graphics-MENU (GUI)	£118
Guide 3 for PC (Hyper Text)	£315
HP New Wave (GUI)	£93
Install (Automatic Installation)	£125
KnowledgePro	£370
Math Coprocessors	CALL
Menuer	£185
Modems	CALL
Phar Lap 386 DOS Ext SDK	£320
PolyAWK v1.3	£111
Thompson AWK Compiler v1.5	£129
RPGII Development System	£985
SCHEMA-PCB (SC-4)	£647
SoundBlaster	£130
Stacker V2.0	£75
System Sleuth (Diagnostics)	£63

PASCAL

Asynch Plus	£107
B-Tree Filer	£85
Graf/Drive Plus Personal	£71
POWER TOOLS PLUS	£84
Topaz	£57
Turbo Analyst	£980
Turbo Pascal for Windows	£102
Turbo Pascal Professional	£138
Turbo Professional	£80

SCIENCE & ENGINEERING

ChiWriter Chemistry Support	£40
Derive v2.0	£161
InGraf	£232
Labtech Notebook	£706
Mat for 286	£199
MathCad for Windows	£267
Mathematics for Windows	£754
PlotLib	£166
SSP/PC Scientific Subroutine Library	£216
Tech*Graph*Pad	£295
Zortech C++ Science & Engineering	£432

SCREEN DESIGN & MANAGEMENT

C Windows Toolkit	£76
Flash-up	£61
Hi-Screen XI	£98
POWER SCREEN	£84
ProPak	£364
Say What?	£29
Vitamin C	£224
WNDX GUI Development System	£283

TRANSLATORS

BAS_C Commercial	£213
FOR_C++.	£767
FOR_C with Runtime Source	£554
Pascal to C Translator	£123
Turbo Plus	£122

UNIX/XENIX

Architect Series Workstation Developer.	£1,347
CorelDraw for UNIX	£449
DataFlex for UNIX 386	£838
Informix PSQL/C Development System v4	£311
LPI COBOL Compiler	£949
Norton Utilities for System V v3.2	£153
Oracle UNIX/XENIX 386	£1,302
SCO UNIX System V/386 Development System	£565
VEDIT PLUS for UNIX/XENIX	£162

WINDOWS DEVELOPMENT

Asymetrix ToolBook	£247
Becker Tools	£64
C++/Views v1.1	£257
Microsoft C with Windows SDK	£363
MS Windows Device Driver Development Kit	£275
Microsoft Windows Productivity Pack	£33
Object Vision	£76
Whitewater Resource Toolkit	£99
WIN++	£141
WindowsMaker Professional	£658
WinSleuth	£63

We will match any advertised Price!

- Prices exclusive of V.A.T. & shipping.
- All major credit cards accepted.
- 30 days invoicing for qualified organisations.
- Special discounts for large or corporate buyers.
- Prices subject to change without notice.

.EXE/2/92

USA ★ SOFTWARE

7-11 KENSINGTON HIGH STREET ● LONDON W8 5NP
Phone Nos.
SALES: 071-938 4800 ● SWITCHBOARD: 071 938 2161 ● FAX: 071 937 5776

Over 3,500 products carried. Call today for price list-disk.

CIRCLE NO. 468

Unlike more complex editors, QEdit only uses a single executable (47 KB) so you can stick it onto a single floppy disk and run it on any machine you're using. You can register your copy of QEdit directly to SemWare in the US by sending $54.95 (+$10 shipping). However QEdit is also distributed in the UK (see Figure 5 for prices). A TSR and OS/2 version of QEdit is also available.

Ed

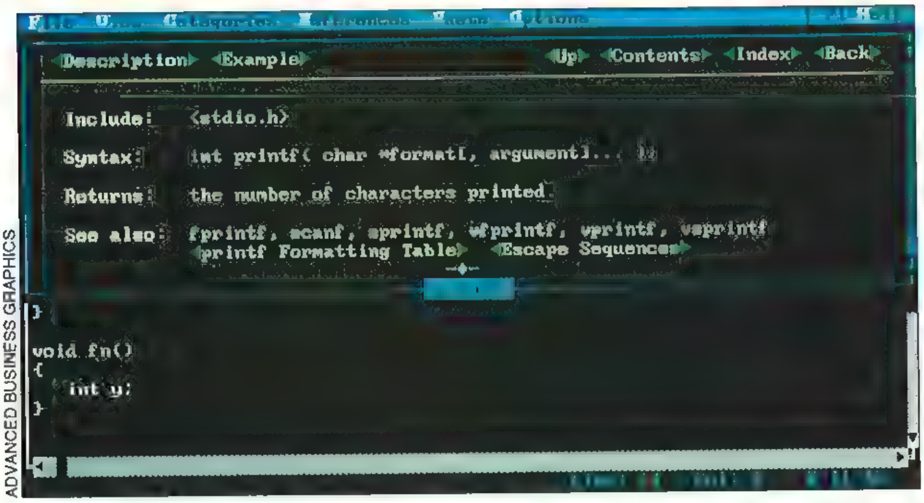
You probably haven't heard about Ed yet, so I'll just put you in the picture. Ed began its life a long time ago (pre-CP/M) and was first ported to CP/M 80 and then on to DOS. In early 1990, the Australian company, Soft As It Gets, released a commercial version.

There is a Status Bar at the top of the screen and a Function Key template at the bottom. This template shows what action each function key (F1 to F10) performs. A pull-down menu can also be activated which gives the short-cut key strokes. There's also a context-sensitive Help system and online documentation. Standard editing functions like highlighting a rows or column of text are, of course, present. Ed supports extremely large files and utilises a virtual memory mechanism for swapping sections of a large file from memory to disk and vice-versa. There can be up to eight windows on the screen at any one time and each window may be created by splitting the screen either horizontally or vertically. Files may be loaded in edit-mode or view-only, enabling the sharing of files over a LAN.

One of the cleverest features of Brief is its ability to work with third-party software

```
Description  Example                    Up  Contents  Index  Back
Include:   <stdio.h>
Syntax:    int printf( char *format[, argument]... );
Returns:   the number of characters printed.
     printf Formatting Table  Escape Sequences
void fn()
{
   int u;
}
```

Figure 2 - Epsilon's Concurrent DOS session



```
Description  Example                    Up  Contents  Index  Back

 Include:    <stdio.h>

 Syntax:     int printf( char *format[, argument]... );

 Returns:    the number of characters printed.

 See also:   fprintf, scanf, sprintf, vfprintf, vprintf, vsprintf
             printf Formatting Table   Escape Sequences

}

void fn()
{
  int u;
}
```

ADVANCED BUSINESS GRAPHICS

Figure 3 - Context-sensitive Help in Brief

To get the most satisfaction from using Ed, you have to write C/C++ source. It directly supports Zortech C/C++, Turbo C/C++ and Microsoft C compilers and there are several built-in utilities which will ease the burden of writing code. First there's C-style indentation, parenthesis-matching and a handy line commenting macro. Then there is the template feature. Now this is a real time-saver. Templates work by replacing a single character by a complete C construct eg <ESC>s<ESC> produces the following :

```
switch (   )
{
    case :
        break;

    default :
        break;
}
```

Do you ever need to quickly look up the definition of an external function that you're using? CTAG is a program that keeps a record of C function call definitions (it also works with non-inline C++ member functions). Once all your files have been processed by CTAG, you can simply view the file containing the function definition by invoking Ed's Edit File menu option, and specifying the name of the function you wish to see. Ed will automatically load the relevant file and place the cursor on the first line of the function's definition. Ed even provides you with a function browser for moving to a given function definition within the file that you're currently editing.

There are also a few pop-up utilities bundled with ED. These include a nifty calculator that can convert from decimal to binary or hex. There's a built-in ASCII table and a table showing the C precedence rules. Although I have concentrated on ED's C programming features, I must point out that it also supports Pascal, Assembler, Modula-2, Clipper and dBASE. ED is a feature-packed editor, filled to the brim with clever ideas for writing code faster, and more efficiently.

Epsilon

Although all the Programmer's Editors in this article were tested under DOS, there are times when you'd like to use your favourite editor on a different platform (like OS/2 or UNIX). As you already know, all UNIX machines are guaranteed to have a copy of `vi`. `vi` is even available under DOS and OS/2. However, I can't see many DOS programmers moving over to `vi`! Epsilon (from Lugaru) is available for DOS, OS/2, SCO UNIX and Interactive 386/ix, providing true multi-platform editing. Unfortunately this limits its capabilities, since many UNIX terminals lack even the most basic cursor controls and text highlighting facilities.

Escape with C-scape

ELEGANT GRAPHICS AND TEXT

C-scape Features

- Graphics — Combine high-resolution graphics with text or menus.
- Object-oriented — Add features and create re-usable code modules.
- Mouse — Use any standard mouse for fast screen control.
- Portability — Write hardware independent code. Supports MSDOS, OS/2, UNIX and VMS. Auto-detects: Hercules, CGA, EGA, VGA.
- Text Editing — Create a full-featured text editor or pop-up note pad.
- Field flexibility — Create masked, protected and marked fields with complete data validation. Use time, date, money, pop-up list and many more functions — or create your own.
- Windows — Choose from pop-up, tiled, bordered and exploding windows with size and numbers limited only by RAM.
- Menus — Choose from pop-up, pull-down, 123-style or slug menus — or create your own.
- Context sensitive help — Link help messages to individual screens or fields. Cross reference messages to create hypertext-like help.
- Screen design — Build any type of screen or form with Look and Feel™ screen designer, which will then automatically convert to C.
- Screen flexibility — Call screens from files at run-time or link them.

The **C-scape**® Interface Management System frees C programmers from the tedium of coding windows, menus, data validation, help and text editing functions.

C-scape is a pleasure to use. With **C-scape**'s object oriented design you can build more flexible, more functional, more portable and unique applications — and you will enjoy doing it.

The Industry Standard.
Many thousands of programmers have given up home-grown libraries and cumbersome, inflexible products for the ease of **C-scape**. In the US, *IEEE Computer* said "**C-scape** is by far the best . . . a joy to use." *PC Magazine* chose **C-scape** to produce its Laboratory Benchmark Series 5.0 software because **C-scape** offers mouse support. Moreover, **C-scape** simultaneously combines text and graphics.

C-scape, from LIANT SOFTWARE CORPORATION, is built around an open architecture, so it can be used with data management or other C libraries. At Systemstar, we offer **C-scape** with **Raima Data Manager** from Raima Corporation to provide a complete development environment.

To port your application from MSDOS or OS/2 to UNIX or VMS, just recompile.

Source code is included in the price and there are NO RUN-TIME ROYALTIES.

For more information about C-scape call Systemstar on (0992) 500919.

SYSTEMSTAR
LIMITED
1-3 Parliament Square, Hertford, SG14 1EX
Telephone: (0992) 500919 Facsimile: (0992) 554261

CIRCLE NO. 469

THE INSTITUTION OF ANALYSTS & PROGRAMMERS

The Institution of Analysts & Programmers represents an elite body of men and women who are leaders of the computing profession. These are people whose expertise enables them to analyse the problems of modern industry, and apply computers to their solution.

Membership of the Institution is a recognised mark of professional status. Designatory letters, which members are entitled to use, indicate their grade within the Institution, and their standing within the profession. Grading depends on age, experience and academic attainment.

Applications are welcomed from all men and women who are engaged in systems analysis or computer programming, or who are training for the profession. Enquiries may be made by letter, telephone or fax.

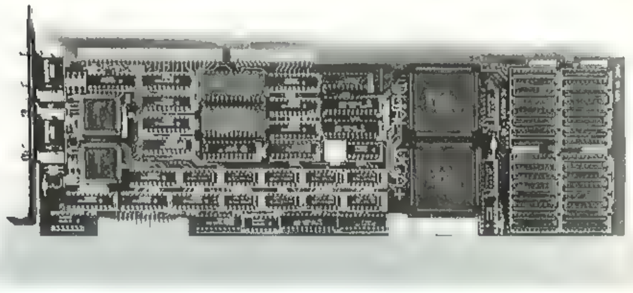
Telephone	Fax and Messages
081-567 2118	081-567 4379

The Institution of Analysts & Programmers
Charles House, 36 Culmington Road,
London W13 9NH. England

CIRCLE NO. 470

MULTIPLE INDEPENDENT MONITORS CAN NOW BE DRIVEN SIMULTANEOUSLY FROM A SINGLE PC OR PS/2

THE DUAL VGA PLUS



- ★ UP TO 800×600 RESOLUTION
- ★ TWO INDEPENDENT SIMULTANEOUS DISPLAYS WITH OVERLAY CAPABILITY
- ★ 100% COMPATIBLE WITH IBM VGA/EGA/CGA/MDA VIDEO STANDARDS
- ★ 4 BOARDS IN ONE SYSTEM GIVE 8 SIMULTANEOUS DISPLAYS

NEW DRIVER! ALLOWS WINDOWS 3.0 TO BE EXPANDED ACROSS FOUR MONITORS

CEBRA COMMUNICATIONS LIMITED
26 LORNE PARK ROAD, BOURNMOUTH
TEL 0202 299048 FAX 0202 299192

CIRCLE NO. 471

Epsilon uses EMACS-style commands, although it is highly configurable. It has a status bar which shows the filename, type of buffer and the percentage of the buffer that has been scrolled down.

In addition to the standard text buffer, Epsilon supports a further three special buffer types - each of these may be displayed in its own window. First there is the `bufed` buffer which holds a list of buffers together with the files that are associated with them. `bufed` operates in a similar way to how the buffer list-box works on other editors. Scrolling through the list enables you to select the current buffer.

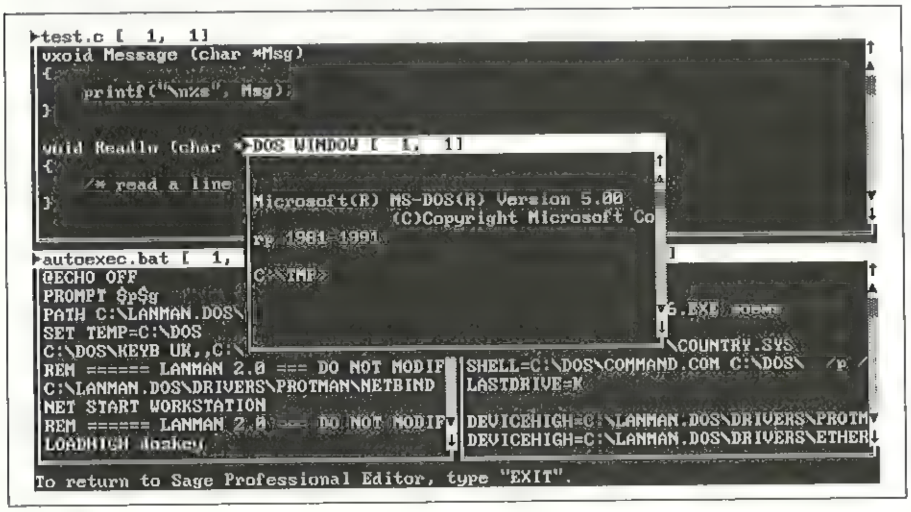
Second there's the `dired` buffer which contains a listing of the current directory. It is possible to select files for deletion using the 'D' key to place a flag on the files you want to delete. The 'X' key actually removes these files, after verifying that this is really what you want. Pressing <Enter> on a given file causes the editor to load the chosen file into the current window.

Next there is the C buffer type. When a C program is loaded into a C buffer, the editor produces C-style indentation and parenthesis/bracket matching. Compiling the contents of a C buffer is a rather involved procedure. This is because Epsilon defaults to the UNIX `make` command. To use Microsoft C you have to pass a dummy parameter to the Epsilon `make` macro which causes the macro to prompt you for a command-line. Compilation errors are trapped by Epsilon and you can step through them using <CTRL>X <CTRL>N. Epsilon also provides a `tag` facility for cross-referencing functions. Running this command on all C source files produces a list of function definitions with filenames and line numbers. However, it is not possible to select a function to read into a buffer - you have to load the file containing the function definition manually, and 'cursor-down' to the relevant bit of code. This 'tag' file is like a normal text file and can be opened any time.

SPE will even run Windows Enhanced mode from within the DOS window

Finally there is the `process` buffer. A concurrent task runs in this buffer, and what's more, it even works under DOS (see Figure 2) - Lugaru claims that Epsilon is the only DOS editor to support concurrent processing (although under DOS, Epsilon will only support one concurrent task). The main problem with the `process` buffer is that it doesn't leave enough memory in the DOS shell to run anything useful - on my machine, a typical concurrent editing session left about 150 KB for DOS (ie not enough to run a compiler or a linker). To get around this, Epsilon offers an 'Exit to DOS' command which needs only 6 KB.

I have never used EMACS before, so most of the keyboard sequences came as quite a shock. This is made worse by the fact that there isn't a menu system, so you have to remember dozens of keystrokes. Luckily the Epsilon manual is well-presented and easy to use. There is online help, but I don't recommend you use it, since it is extremely tedious. Epsilon is a fine editor, although it lacks some of the more 'user friendly' features found on most DOS editors.



```
▸test.c [   1,   1]
 vxoid Message (char *Msg)
 {
     printf("\n%s", Msg);
 }

 void Readln (char ▸DOS WINDOW [   1,    1]
 {
     /* read a line
 }                   Microsoft(R) MS-DOS(R) Version 5.00
                                  (C)Copyright Microsoft Co
                     rp 1981-1991

▸autoexec.bat [   1,                                       ]
 @ECHO OFF           C:\TMP>
 PROMPT $p$g
 PATH C:\LANMAN.DOS\                                  6.EXE
 SET TEMP=C:\DOS
 C:\DOS\KEYB UK,,C:\                                 \COUNTRY.SYS
 REM ====== LANMAN 2.0 === DO NOT MODIF SHELL=C:\DOS\COMMAND.COM C:\DOS\  /p /
 C:\LANMAN.DOS\DRIVERS\PROTMAN\NETBIND  LASTDRIVE=K
 NET START WORKSTATION
 REM ====== LANMAN 2.0 === DO NOT MODIF DEVICEHIGH=C:\LANMAN.DOS\DRIVERS\PROTM
 LOADHIGH doskey                        DEVICEHIGH=C:\LANMAN.DOS\DRIVERS\ETHER
To return to Sage Professional Editor, type "EXIT".
```

Figure 4 - Overlapping Windows in SPE

Brief

I've been reliably informed that Brief is one of the most popular Programmer's Editors for DOS and OS/2. So, why do people buy it? What do you get for two hundred quid? (Brief costs nearly as much as a good C compiler.)

Brief is supplied with a lot of documentation - over 600 pages in total, split into a user manual and a macro reference manual. There is also online Help and a context-sensitive Help system. The Help system is easy to use and provides useful info on most of Brief's features, including help on how to manipulate Blocks, Buffers, cursor controls, windows and how to perform search and replace.

There is a setup program to reconfigure Brief after it has been installed. This automatically makes changes to your `CONFIG.SYS` and `autoexec.bat` files. In addition to being able to configure the screen colours, line length, number of levels of undo etc, the setup program allows you to provide a command-line for compilation inside Brief.

Brief supports several languages including Ada, Basic, C, Cobol, Fortran, Modula-2 and Pascal. <ALT> F10 compiles the current buffer and produces a list of compilation errors in an error list-box, activated with <CTRL> P. It is possible to select which compiler to run, although this can only be done in the setup program. The file extension enables Brief to identify which compiler to invoke when <ALT> F10 is pressed (ie a .C file is compiled with CL, a .asm file is assembled with MASM). Brief's support for C, provides the programmer with a template facility (like Ed's) to reduce typing. Unlike Ed, this only requires a single character followed by a space to invoke the template macro. For instance the key sequence <i> <space> produces the following:

```
if ()
```

One of the cleverest features of Brief is its ability to work with third-party software. Brief has built-in support for the PVCS, TLib and Sourcer's Apprentice version control system, although I was not able to test this personally. There is also support for the Microsoft Programmer's Workbench (PWB) and Microsoft's Quick Help (supplied with Microsoft C V6.0). Quick Help provides context-sensitive Help on C syntax and standard functions. It's very neat.

F77L–EM/32 & Lahey Ergo OS/386

Port mainframe programs as large as 96MB to 386/486's with this 32-bit DOS-Extender compiler. The winner of PC Magazine's 1988 Technical Excellence Award just got better. New Version 4.0 includes: Programming Tools, Popular Fortran 90 features, Virtual Memory Support, DESQView support, New Documentation and Free Unlimited Runtime Licenses. F77L-EM/32 and OS/386. *£875.00 plus VAT*

F77L

The fastest real-mode compiler available. F77L can take advantage of your 386 PC by generating 32-bit instructions. New Version 5.0 includes: Fortran 90 features, Weitek support, and Video Graphics. *£375.00 plus VAT*

Lahey Personal Fortran 77

Version 3.0: Full ANSI 77, Editor, Debugger, Linker, Library Manager, Microsoft and Borland C interfaces. A great learning tool at an unbeatable price. *£79.00 plus VAT.*

When people talk about FORTRAN
the name mentioned most often is
Lahey

System Science

3-5 Cynthia Street, London N1 9JF
Tel: (071) 833 1022 Fax: (081) 837 6411

CIRCLE NO. 472

CopyControl

THE NEW GENERATION OF COPY PROTECTION

NO	User hassle
NO	Back-up problems
NO	Hardware add-ons or special disks needed
NO	Changes to source code required
YES	CopyControl beats ALL bit-copier Programs
YES	Floppy disks, hard disks and networks supported
YES	CopyControl is totally transparent to the user
YES	You can limit program use by no. of copies, no. of executions or date
YES	CopyControl works on all IBM compatibles
YES	Free demo disk available

For Further Information Phone or Write to

microcosm

Microcosm Limited
17 Cranbrook Road, Bristol BS6 7BL
Telephone: 0272-441230 Fax: 0272-427295

CIRCLE NO. 473

JPI

the compiler people

Imagine a powerful integrated development environment common to whichever languages you choose, capable of supporting DOS, Windows or OS/2 development embodying the latest in OOP technology – it's called ***TopSpeed*** from JPI. Its modular architecture opens up new realms of choice – simply slot-in languages, source libraries, and toolkits to build a development system that meets your needs. No redundant components – what you want is what you get!

***TopSpeed* Environment** – Multi-window editor, powerful Project system, Hypertext help (environment, all languages and library), debugger, syntax checker, menu or command line driven, EMS support, pop-up calc, plus . . . plus . . . DOS or OS/2 **£59**

***TopSpeed* TechKit** – For power programming. Supports DOS DLLs, post mortem debugging, advanced overlay manager, TopSpeed assembler, .EXE file compressor, .OBJ file disassembler. **Windows 3 resource compiler**, Windows 3 run-time libraries, plus . . . plus . . . **£59**

***TopSpeed* C++** – makes C++ lean and mean Unique to JPI – TopSpeed C is the only true AT&T 2.1 C++. TopSpeed C++ gives you SmartMethod® Linking which eliminates unreferenced classes, methods and even virtual methods – a real breakthrough for OOP programming. Includes short based pointers, concurrent tasking – even with DOS, plus . . . plus . . . DOS or OS/2 **£59**

Rogue C++ Class Library **£59** DOS or OS/2

***TopSpeed* C** – *the standard is enhanced.* The only ANSI certified C. Generates compact high quality code for DOS, OS/ 2 and Windows 3. Includes run-time error checking, multi-threading, mixed memory models, links to C++, Modula-2 and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Modula-2** – the world leading implementation of Modula-2. This strongly typed and highly structured language includes type safe conversion between objects, OOP extensions with true multiple inheritance, virtual pointers, based pointers, links to C, C++ and Pascal, plus . . . DOS or OS/2 **£59**

***TopSpeed* Pascal** – the next generation. Power-up your Pascal, convert to TopSpeed, then tune it up with the hottest compiler for DOS, Extended Dos, OS/2, or Windows. ISO 7185 conformant, Turbo to TopSpeed converter, ISO conformant arrays, dynamic strings, separate compilation units, OOP extensions plus much more . . . DOS or OS/2 **£59**

Library Source Kits available in all languages **£59**

***NEW* *TopSpeed* DOS Extender**

Power-up your programming with JPI's own DOS Extender toolkit. Blast the 640K DOS barrier and supercharge your applications. Built-in disk based virtual memory management system for code *and* data, multi-tasking support using OS/2-like threads and a pre-emptive scheduler, OS/2 format DLLs supported, automatically loaded and unloaded on demand, plus . . . Royalty free. Full source code available.

Call JPI on (0234) 267500 now for your free copy of the TopSpeed 1991 Compiler Catalogue

TopSpeed products are available from:
GreyMatter 0364-53499 · System Science 071-833-1022
RTA 081-656-7333

JPI CIRCLE NO. 474

3 The Mansards, Tavistock Street, Bedford MK40 2RX
Fax: (0234) 217094

TopSpeed C++ TopSpeed Modula-2 TopSpeed Pascal TopSpeed C

Developing software on a LAN is fast, efficient, and dangerous.

We make it safe, controllable and auditable.

Experienced managers are rightly concerned about the integrity of LAN development environments. It's too easy to overwrite or corrupt files.

To safely develop on a LAN you need control over file changes, the ability to recover any module revision or system version, audit trails, and security that doesn't hamper productivity.

The solution: our LAN development environment, three tools that operate independently or snap together synergistically.

The primary element is PVCS Version Manager which provides version control and change management features, files and function security, automatic audit trail and detailed project reports.

The best way to access PVCS Version Manager is via our PVCS Professional Editor, an extensible control centre that simplifies access to any tool and provides industry strength editing. PVCS Professional Editor invokes PVCS Version Manager automatically. Just open modules to edit and the PVCS Version Manager window pops open.

The final element in our LAN development environment is PVCS Configuration Builder - a sophisticated tool for automatically performing correct builds. A single command recompiles all modules that have been changed, re-links as necessary, and produces a correct, conforming program.

Don't Let These Special Offers Pass You By.

Special offer prices are valid until 29th February 1992

	DOS or OS2	DOS and OS
PVCS Version Manager	£315	£416
PVCS Configuration Builder	£136	£180
PVCS Professional Editor	£160	£208
Mainframe Gateway	£112	--

Software Express · Portmill House, Portmill Lane, Hitchin, Herts SG5 1DJ. Tel: 0462 422525

CIRCLE NO. 475

	QEdit	Ed	Epsilon	Brief	PVCS
Grey Matter (0364 53499)	£45	-	£190	£199	£195
QBS Software (081 9944842)	-	£109	-	-	-
Software Construction Company (0763 2441141)	£39	-	£165	£199	£195
Software Paradise (0222 877521)	£65	-	£164	£191	-
System Science (071 8331022)	£45	-	£175	£195	£189

(Please note that all prices include p&p in the UK, but not VAT)

Figure 5 - Editor Prices

You only have to run Brief's qh macro to get help on a keyword/function at the current cursor position (see Figure 3). All of these features make Brief a brilliant editor, albeit with a hefty price-tag.

PVCS Professional Editor

As is the case with Epsilon and Brief, the PVCS Professional Editor (also called the Sage Professional Editor - SPE) is available under DOS and OS/2.

SPE provides all the usual cursor controls and editing commands, although it doesn't provide a column copy facility. There is a menu-bar which is activated with <CTRL> H, and a status bar that shows the filename. There is a Help system which can be activated from the menu or by pressing <F10>. Although not context-sensitive, the online documentation is well presented. Help is arranged as two scrollable windows placed side-by-side. The leftmost window contains a list of SPE's commands. As you scroll down this window, the right window displays documentation on each topic.

There is a DOS window, like Epsilon's, although it is not concurrent. The DOS windows uses only 24 KB of memory, so you should be able to run a compiler inside it. SPE will even run Windows Enhanced mode from within the DOS window. SPE also has a DOS shell, and it is able to compile a buffer from within the editor. On typing <ALT>F10, SPE pops up a window containing an editable commandline.

When compiling a buffer, SPE produces an error window and positions the cursor on the line containing the first error. It is possible to step through each error in turn, however SPE also enables you to quickly move to a particular error by scrolling down the error window.

SPE provides built-in support for C. It is able to produce C-style indentation and there is also a template facility. SPE can emulate Brief, vi, WordStar and MsWord and there is also a keyboard macro recorder. There's a built-in ASCII table which lets you embed any ASCII character into your text by entering its ASCII value (<ALT> + number from numerical key-pad is quicker if you know what you're doing). SPE is the only editor in this article that supports multiple overlapping windows. In addition to being able to create windows side-by-side, SPE allows windows to be placed on top of each other (see Figure 4). When you have several files opened at the same time, arranging each buffer as an overlapping window provides a far more intuitive way of editing multiple files than using a single window with a buffer-list. It also uses less screen space than a tiled-window arrangement - at least you won't have to resort to excessive scrolling in order to do anything useful.

Conclusion

Choice of editor is almost always influenced by past experienced. If you're a die-hard fan of WordStar, I won't waste space convincing you that there are better things available. However, most of the editors in this article provide emulations for the more popular text editors (including WordStar). So which one should you buy? Well, it all depends on what you intend to use it for. Qedit is an ideal replacement for MS DOS 5.0's `edit` - it provides far more functionality and it doesn't crash. Epsilon offers a terse, UNIX-style environment and is available under UNIX as well as DOS and OS/2. Brief and SPE are similar products, targeted at the same market (ie C programmer's looking for a high-end professional editor). My favourite? It has to be Ed. It offers many of the features found on Brief or SPE - but at almost half the price.

EXE

QEdit, Epsilon, Brief and the PVCS Professional Editor were supplied by System Science on 071 8331022. Ed was supplied by QBS on 081 9944842.

Wasting our time

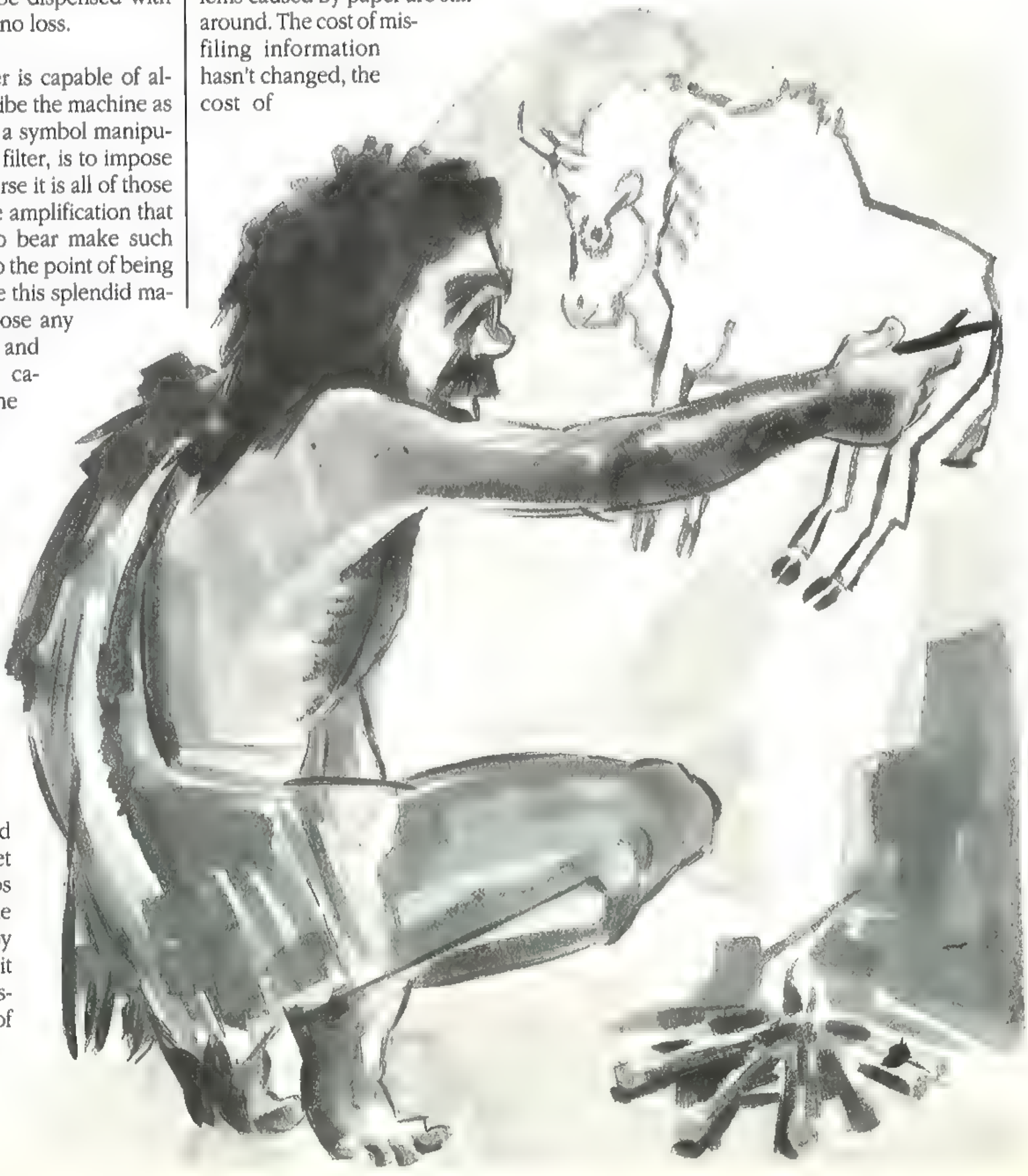
Why do we buy computers and write programs? Jules thought he knew, but is not so sure.

It was during an argument when the thought struck me. It was such a surprise that I conceded the point of the argument, in order to go away to consider it! The thought was this; computers don't actually do anything. I don't mean this in a literary or figurative sense, nor am I talking about abstract philosophy. I mean it quite literally; most computers could be dispensed with entirely with practically no loss.

In principle, a computer is capable of almost anything. To describe the machine as a bunch of switches, or a symbol manipulator, or an information filter, is to impose restrictions on it. Of course it is all of those things, but the immense amplification that computers can bring to bear make such descriptions simplistic to the point of being misleading. We can take this splendid machine, and we can impose any concept on it we wish, and the computer will be capable of responding to the imposition and offering its user a clean, effective metaphor.

Most programs which have been written use the same metaphor - paper. I suppose this makes sense - everyone who can type can read, and most of what we read is made of paper. When computers were first manufactured in the '50s they were set to work automating jobs which were being done perfectly adequately by armies of clerks, albeit slowly. Nobody questioned why the armies of clerks were necessary, and nobody questioned that the same work needed doing on the computer. It wasn't until Marks and Spencer worked out how much it cost to misfile a piece of paper that they decided they could throw some of it out - 120 tons of it in fact!

As long as computers are automating work that originated on paper, all the old problems caused by paper are still around. The cost of misfiling information hasn't changed, the cost of getting the data onto the forms in the computer hasn't changed, and the cost of someone wading through acres of figures hasn't changed either. The cost of processing the data has decreased, of course, which has permitted yet more paper to be produced. Tasks which used to use paper out of necessity, like fin-

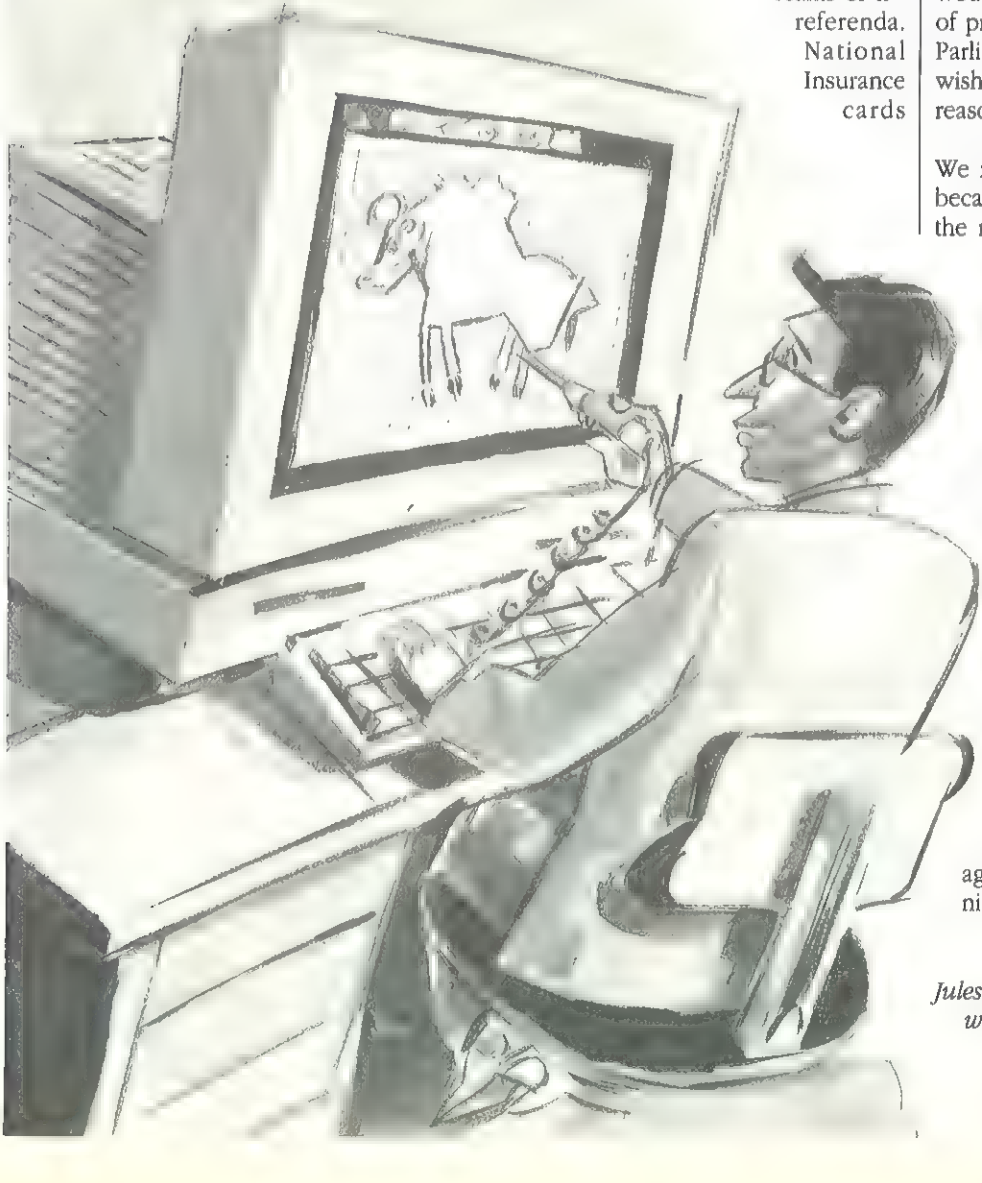


ance, CAD, and graphics, instead of being freed by computers, have been computerised monolithically, paper and all. Now, the wonderful world of Windows is opening up before us - here is computing like you've never seen it before. Ha! What is the central metaphor of windowing systems? Paper, that's what!

'Aha!' I hear you say, 'but most computers are made for business, and business needs accurate records, doesn't it? Paper is good for that!' Well, yes and no. Most people who run businesses do so because they want to build something, not because they like paperwork. I have yet to meet an executive who doesn't complain about his paperwork, no matter how many computers and secretaries he is given. The only reason an exec needs to know his company's turnover correct to three pence is because the taxman demands it, but whv does the taxman need such precision? The exec doesn't work out his buying strategy to the nearest three pence, does he? Why isn't 1% precision good enough all round?

Consider, as an alternative, the spectacular strides which have been made in musical instrument design. One can get cheap machines which can synthesise practically any noise one can imagine, and the proof is that often synthetic instruments sound better than their acoustic counterparts. There are computers into which you can hum a tune, and they can, in real time, figure out harmonies and rhythms to back you up. Almost every band now comprises a guitarist, a drummer, and a programmer! Furthermore, these machines are usable by musicians - their buttons and knobs are annotated in terms a musician can understand.

For another example, let us pursue the idea of removing paper from a process which traditionally demands reams of it - referenda. National Insurance cards are now plastic, with a magnetic stripe on the back. Suppose every post office in the country is equipped with a terminal, into which you insert your card, and are shown a list of the issues being debated in parliament. Instead of politicians having to guess whether we want to be European, or whether we want a bill of rights, or whether we want to re-introduce Sunday trading, we can tell them. The computers could keep a record of what issues a citizen has voted on (though probably not the vote he has cast) to avoid repetitions. We could build the first real democracy the world has ever seen!



This is looking good, let's follow it further. One could subscribe from one's telephone, so it is not necessary to stand in queues. Instead of seeing politicians rubbishing each other on the telly, we could augment the system with statements of position from interested parties. One could even contribute one's own statement, which would become more easily accessible as more people looked at it. A citizen would be able to introduce his own bill, which again, would work its way up the system to a place of prominence if interest in it demanded. Parliament would then have to follow the wishes of the people, or else give a good reason why it won't.

We need representatives at the moment, because calling a referendum - the paper, the manpower, and the officialdom - all costs money. Here is a system which could slash those costs, and improve the service to boot, because what referenda are for, in the final analysis, is to take decisions. The engineering is almost trivial, and the technology is tried and tested.

Returning to my original assertion, it seems to me we are wasting some wonderful technology. Computers are not new (they we first built in 1943) and are largely mature by now. We have enough knowledge and skill now to build things to pretty good precision (which is good enough), and the only thing which is blocking the technology from being really useful is conservatism and lack of imagination. I'm afraid most of us are spinning our wheels.

EXE

Jules May is a consultant who feels naked without a computer in front of him, and is currently trying to find the time to write lots of new programs. He can be contacted on 0707 44185, or on CIX as jules.

Segments From Hell

Once you've navigated the treacherous waters of the memory management 'River Styx', you too can give your DLLs a touch of class. Laine Stump puts a C++ class library into a Windows DLL.

Windows Dynamic Link Libraries (DLLs) seem to be the perfect package for implementing and distributing class libraries; rather than bloating all those EXE files with the same library code, you can build a DLL and call the same code from any EXE. This not only saves space on disk, it (more importantly) saves memory at runtime.

But putting your handcrafted classes into a DLL isn't as simple as just recompiling. Besides making sure that the proper names are exported from the DLL and imported to the application (APP), you must take care to avoid problems with the 'Dual Heap' structure of the DLL-APP pair - not only are the code of the DLL and APP in separate segments, they each have a separate automatic data segment (heap, or DS) as well. (Figure 1 shows a simplistic picture of the segments of a DLL and two APPs; note that an APP's stack is in the same segment as its heap, while a DLL has no stack - it borrows the stack of the calling APP).

With all these complications, it's not difficult to construct a program where inheritance, virtual functions, and static data members, (among other features) don't work at all properly.

As usual, the explanations and examples included with compiler packages do not exercise all these capabilities (BC++'s DLL example had a class, but did not derive any other class from it; Zortech's DLL example was written in straight C), so I set to explore the subject by creating my own example program, with APP classes derived from DLL classes, static data members, and virtual functions called both from the APP and from the DLL. Along the way, I recorded the pitfalls, problems, hints, and solutions I found, and present them here so that the rest of you won't have to go through the same torture (fun torture though it was).

Piece

The example I will use to help in my explanations is a simple class called Piece. A Piece is any object that can be displayed in a window. The base class has functions for hiding, showing, and moving a Piece, as well as a virtual function called Paint() which is redefined for each derived class. Piece resides in a DLL, along with a derived class called cRectangle. PIECE.DLL needs PIECE.CPP, PIECE. These files are shown in Figures 3, 4 and 5.

The main program, PieceTst (see Figures 6 and 7), derives a second class, cEllipse, from Piece, and creates three cRectangles and two cEllipses which it stores in an array of Piece *. WndProc works together with members of Piece to allow the user to drag these pieces around the screen with the mouse. Except for Piece and its derivatives, the rest of the program is in standard C style, as I didn't want to clutter up the discussion with more than one set of classes. Ideally the Pieces would be stored in some type of DisplayList, and the windows and even the application itself would also be instances of C++ classes. I leave those as exercises for the hapless reader.

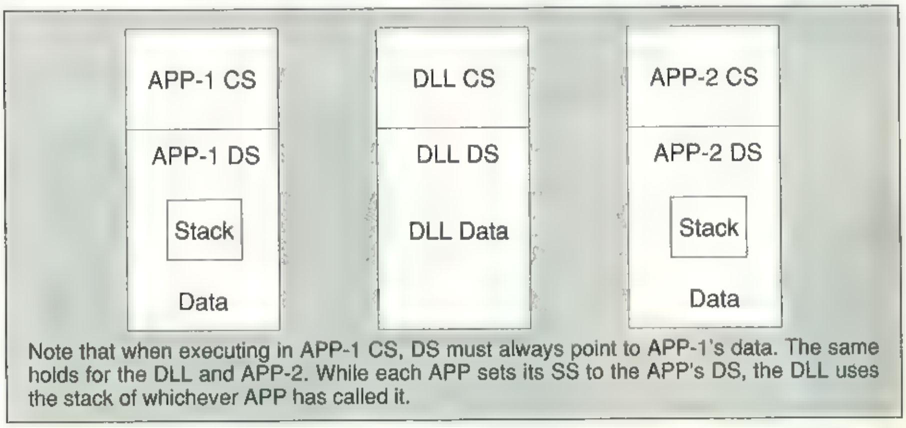
PIECETST.EXE is created from PIECETST.CPP, PIECE.H, PIECETST.DEF, PIECE.LIB (created by running IMPLIB on PIECE.DLL) and PIECETST.RES.

I developed and tested Piece and PieceTst using Zortech C++ v3.0r4 (ZTC); the necessary commands to compile are in Figure 2 (M.BAT). Only small changes should be needed for BC++ v3.0 - different compiler switches and a slightly different method of exporting functions from the DLL.

EXPORTing from the DLL

The names of all functions in a DLL which will be called from APPs and other DLLs must be available for load-time linking. Windows calls this 'exporting' the function. The client program then 'imports' the function to use it.

You can export a function in either of two ways: list it in the EXPORTS section of the DLL's .DEF file, or tag the function definition with the keyword _export. The former method makes for impressive looking lists, but can be quite awkward, as the name listed in the .DEF file must be the function's mangled name.



Note that when executing in APP-1 CS, DS must always point to APP-1's data. The same holds for the DLL and APP-2. While each APP sets its SS to the APP's DS, the DLL uses the stack of whichever APP has called it.

Figure 1 - Segments of two APPs and a DLL.

THE SOFTWARE CONSTRUCTION COMPANY

BORLAND

Borland C++	£199
C++/App. Framework	£269
Turbo C++ 2nd. Edition	£49
Turbo C++/Turbo Vision	£69
Turbo Debugger & Tools	£85
Turbo Pascal v.6.0	£69
Pascal Professional	£129
Pascal for Windows	£99
ObjectVision	£79
Paradox 3.5	£335
Paradox Multi Pack	£399
Paradox Engine	£199
Paradox SQL Link	£215

UK AUTHORISED

DATABASES

Advanced Revelation	£795
Advanced Revelation -	£695
askSam	£255
askSam LAN	£579
Clarion Personal Dev. 2.0	£79
Clarion Professional Dev.	£679
DataEase	£485
DataEase - LAN 3-PAK	£485
Dataflex	£449
Dataflex Multiuser	£1065
Knowledgeman/2	£645
Magic PC	£325
Open Access III	£449
Open Access III Compiler	£385
R:BASE 3.1	£515
R:BASE Compiler	£190
R:BASE 3.0 LAN Pack	£645

DBASE

Clipper 5.01	£325
dBASE III PLUS	£395
dBASE IV	£495
dBASE IV Developer's Ed.	£695
dBFast/DOS	£255
dBFast/PLUS	£255
dBXL	£165
dBXL LAN	£389
Force	£449
FoxBASE+	£295
FoxBASE+/386	£425
FoxPro	£515
Palcom "Paradox Compiler"	£319
Palcom Network Version	£645
Vulcan	£320

OBJECT-ORIENTATED

Smalltalk/V	£65
Smalltalk/V 286	£129
Smalltalk/V Windows	£325

CASE TOOLS

EasyCASE Plus 3.0	£320
EasyCASE Plus Prof. 3.0	£420
Smart Case	£195
System Developer I	£325

VERSION CONTROL

MKS RCS 5.1	£165
MKS RCS 5.1 - 5 User	£645
PVCS Professional	£319
PVCS - 5 User	£1345
SMS	£395
Sourcerer's Apprentice	£175
Sourcerer's Apprentice	£429

TEXT AND UNIX TOOLS

MKS AWK	£65
MKS AWK DOS & OS/2	£129
MKS Programmer's Platform	£429
MKS Toolkit	£165
PolyAWK Toolkit	£145
SpellCode	£65

TRANSLATORS

BASTOC	£515
BAS_C Commercial	£579
CodeTranslator *NEW*	£129
FOR_C	£359
Pascal to C Translator	£99

EDITORS

BRIEF for DOS & OS/2	£199
BRIEF upgrade	£60
BRIEF with dBRIEF	£275
BTags	£35
Cheetah	£159

Microsoft

BASIC PDS 7.1	£219
Visual BASIC	£99
C 6.0 Compiler	£210
C 6.0 & Windows SDK	£325
COBOL 4.50	£425
FORTRAN 5.1	£199
MASM 6.0	£69
MS-DOS 5 Upgrade	£50
Pascal Compiler	£149
QuickBASIC	£50
QuickC	£50
QuickC/Assembler	£85
QuickC for Windows *NEW*	£99
QuickPascal	£50
Source Profiler	£49
Windows 3.0	£65
Windows S.D.K.	£220

U.K. Authorised

dBRIEF for dBASE/Clipper	£99
dBRIEF for Paradox	£99
dBRIEF for R:BASE	£99
EDT+	£190
EMACS	£210
Epsilon	£129
KEDIT	£99
MKS Vi	£99
Multi Edit Professional	£115
Norton Editor	£50
SPF/PC	£159
SYNDIE	£319
VEDIT PLUS	£120

LINKERS

.RTLink	£190
.RTLink/Plus	£319
ALINK	£129
Blinker	£175
Plink w/LTO	£359

MAKE UTILITIES

MKS Make	£99
MKS Make - 5 User	£320
PVCS Config. Builder	£129
PVCS Config. Builder/5 User	£485

PARSERS AND LINTS

MKS Lex & Yacc	£165
PC-Lint	£90

PROFILERS

.RTLink/Plus	£319
CHARGE	£85
Codesifter	£79
Inside!	£85
PC Metric	£129

DEMO BUILDERS

Dan Bricklin's Demo II	£190
Instant Replay III	£99
Instant Replay Prof.	£385
Show Partner F/X	£259

MEMORY MANAGERS

386 MAX Version 5	£85
Above Disk	£79
Blue Max	£100
HEADROOM	£85
HEADROOM Network Extens	£39
HI386 Complete	£65
MOVE'EM	£59
QEMM-386	£65
QEMM-50/60	£65
QRAM	£55
Turbo EMS	£65

DOCUMENTATORS

4c for Brief	£79
ABC Flowcharter	£190
ASMFLOW	£129
C-Clearly	£85
C-Doc	£120
Clear for C	£129
Clear for dBASE	£129
dAnalyst for C and C++	£190
dAnalyst for Clipper	£190
dAnalyst for dBASE III	£190
dAnalyst for dBASE IV	£190
dAnalyst for FoxBASE +	£190
EasyFlow	£99
Flow Charting III	£165
SourceDoc	£190
The Documentor	£190
Tree Diagrammer	£65

DISASSEMBLERS

Sourcer 486	£85
Sourcer w/BIOS pre-proc.	£110

PROFESSIONAL SHAREWARE FOR DEVELOPERS

ASM Library (20)	£40
C/C++ Library (68)	£66
dBASE/Clipper (87)	£100
DOS Utilities (37)	£40
Novell Netware (54)	£66
Turbo Pascal (38)	£40
Visual BASIC (11)	£40
Windows (64)	£66

(Number of diskettes)

E-MAIL

cc:Mail	£450
cc:Mail EXPAND	£385
cc:Mail Remote	£190
DaVinci eMAIL	£645
DaVinci eMAIL Combo	£965

CODE GENERATORS

DATABOSS	£395
Layout	£195
ObjectVision	£259
ObjectVision for Windows	£259
PRO-C	£259
R&R Code Generator	£99
UI2 Developer's Release	£385
UI2 Touch & Go	£259
Zachary	£259

DEBUGGERS

Periscope 286/386 Pod	£389
Periscope I OK	£319
Periscope I/512K	£385
Periscope I/MC	£449
Periscope I/MC 512K	£449
Periscope II	£145
Periscope II X	£129
Periscope IIw/Switch	£145
Periscope IV (16 MHz)	£775
Periscope IV (25 MHz)	£900
Sherlock	£129
SVS DBG	£210

MULTI-TASKING

DESQview 2.3	£85
DESQview 386	£125
Dr. Switch-OnCall	£65

BATCH ENHANCERS

Batcom	£39
Beyond.BAT	£65
Command Plus	£85
ProKey Plus	£85
The Builder	£99

TERMINAL EMULATORS

BLAST II	£195
COTERM/100	£145
COTERM/220	£160
COTERM/220	£160
COTERM/4010	£165
EM 320	£199
EM 4010	£230
EM 4105	£330
EM 4105 Network	£345
ICE.10	£190
SmarTerm 125	£190
SmarTerm 2392	£65
SmarTerm 240	£225
SmarTerm 320	£129
SmarTerm 340	£225
SmarTerm 400	£115
SmarTerm 4014	£145
Smarterm 470	£190
ZSTEM 100	£65
ZSTEM 220	£99
ZSTEM 240	£190
ZSTEM 4014	£65

SCO UNIX

SCO Operating System-2	£525
SCO Operating System Unltd.	£750
SCO Development System	£875
SCO Open Desktop - Pers.	£895
SCO Open Desktop Dev. Sys.	£1295
SCO XSight Runtime	£350
SCO XSight Dev. Sys.	£350
SCO XSight Complete Sys.	£650
SCO Multiview	£395
SCO VP/ix - 2 user	£450
SCO VP/ix - Unlimited	£850
SCO FoxBase+ 286	£795
SCO FoxBase+ 386	£995

UPGRADES

App. Framework/Borland C++	£96
Turbo Vision for Turbo C++	£46
Turbo C to Borland C++	£100
Any Borland product to C++	£130

CALL FOR OTHER UPGRADES

*All prices exclude V.A.T. U.K. carriage included. *One offer per order at published prices. Offer required must be stated at time of order.* VISA or ACCESS accepted. Credit to approved accounts.

SUMMER '91 CATALOGUE
Call for your free copy.

1 THE MALTINGS, GREEN DRIFT, ROYSTON, HERTS. SG8 5DB
TELEPHONE (0763) 244114 FACSIMILE (0763) 244025

CIRCLE NO. 477

I used the second method in Piece. Notice that each function of the class is preceded by the word EXPORT (#defined to be _export if compiling for a DLL, or null if compiling for a normal static link). The compiler and linker work together to make the mangled name of these functions available for export. Much easier.

You will notice that two types of member functions are not exported with _export: inline functions and destructors. Inline functions are not listed as _export because their code does not exist in the DLL anyway; the function isn't imported, but merely inserted into the APP anywhere it is 'called'.

I couldn't use _export on destructors because of a bug in the current release of ZTC which flags such attempts as an error. Until Zortech makes another release, we're stuck figuring out the name of the destructor and listing it in EXPORTS in the .DEF file (see PIECETST.DEF). Find this name by looking in the linker's .MAP file - destructors always have the name

```
___dt__#classnameFv
```

where # is the number of characters in the class name.

Seasoned Windows programmers should note that the requirement of exporting destructors via the .DEF file means we can't use the 'reduced prolog' compiler switch (-W2) as it only allows functions marked with _export in the source to actually be exported. I used -W1 (full prolog for all far functions) in my tests.

No data members are exported in any manner, as instance data can be located via the this pointer, while static data cannot be accessed directly from outside the DLL anyway, as we'll see momentarily.

IMPORTing to the APP

Once a class's member functions have been exported, the APP must import them in order to use the class. There are two ways of accomplishing this as well: listing the desired function names in the IMPORTS section of the APP's .DEF file (in mangled form, of course), or creating an 'Import Library' from the DLL. As we all hate dealing with mangled member function names, we will again choose the second method, which turns out to be simple to the point of disgusting - run IMPLIB on the DLL, then add the produced .LIB file to the compile command for the APP. Don't forget to rerun IMPLIB any time you add or remove a function from the DLL.

Memory Model

Windows is a weird duck in terms of memory models. The freewheeling cauldron of moveable data segments and discardable code segments is frightening if you stop to think. A remnant of its real mode past, Windows attempts to do 'big boy' memory management tasks purely in software. For this to work correctly, well behaved programs should allow as many of their segments as possible to be moved at almost any instant. You can't allow this and have far pointers at the same time, hence the popularity of Medium model (far code, near data) in Windows programs. Large model is allowed, you just have to lock all your data segments in place (and face lynching by the Charles Petzold fan club).

```
    ztc -mlw -W1 -o -D__DLL__ piece piece.def piecetst.res
    implib piece.lib piece.dll
    ztc -ml -W3 -o piecetst piecetst.def piecetst.res piece.lib
```

Figure 2 - M.BAT - compile command for PIECE and PIECETST

```
#include <windows.h>


#ifdef __DLL__
#define EXPORT  export
#else
#define EXPORT
#endif

enum TIMES {IMMED, DELAY};

class Piece
{ // Graphics "Piece" base class
  // static member data
  static HWND  hCurWnd; // cur. window
  static HDC   hDC;  // device context
  static PRECT Inv;  // invalid region
protected:
  // instance member data
  HWND hWnd; // owner window
  RECT bnd;  // bounding rectangle
  DWORD color;
  WORD visible;


  // "real" paint - defined by all,
  // but only called by other members
  virtual void EXPORT Paint() = 0;
public:
  static void EXPORT
   SetContext(HWND hw, HDC h, PRECT i);
  HDC EXPORT GethDC();


  EXPORT Piece(HWND hw,
               int x, int y, DWORD c);
  virtual ~Piece();
  virtual void EXPORT ReDraw();


  int  GetX() { return bnd.left; }
  int  GetY() { return bnd.top; }
  int  EXPORT Contains(int x, int y);
  WORD EXPORT Hide(WORD when = DELAY);
  WORD EXPORT Show(WORD when = DELAY);
  void EXPORT MoveAbs(int x, int y);
};


// cRectangle - DLL class from Piece
class cRectangle : public Piece
{
public:
  EXPORT cRectangle(HWND hw,
         int x, int y, int dx, int dy,
                             DWORD color);
  ~cRectangle();
  void EXPORT Paint();
};
```

Figure 3 - PIECE.H

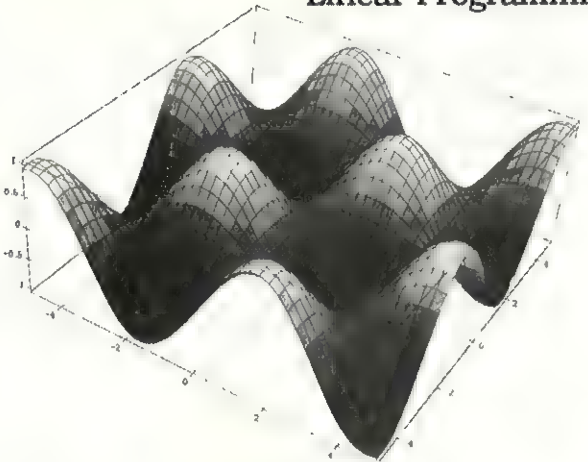
Unfortunately, if you want classes in a DLL, you must compile everything with Large model. I first resisted the idea, but when I learned that even the BC++ OWL library DLLs can only be used with Large model, I resigned myself to fate, and comforted my soul by repeatedly chanting the final section of the Memory Management chapter in Petzold's Windows book, ignoring the part earlier on that referred to Large programs as 'pigs'.

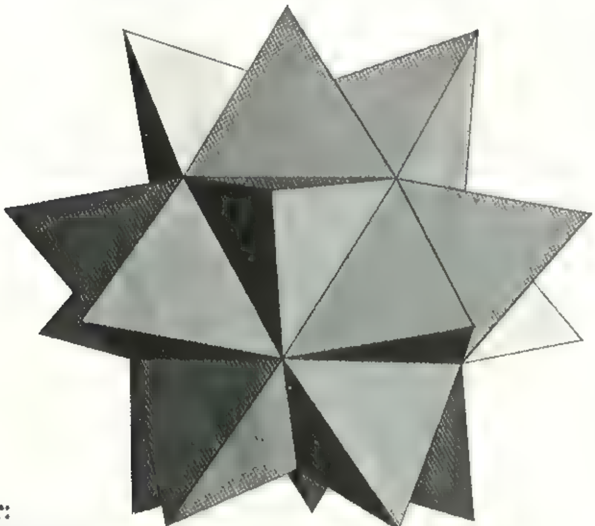
Although there are many, I will tell you two reasons why any program having classes in a DLL must be compiled as Large model (sorry, Charlie). The cause of both is that the DS of the APP and the DLL are separate, and an object could be located in either one:

1) To allow access to objects in either DS, the this pointer of all member functions must be far. It is reasonable for a member function to call Windows, and every Windows programmer knows that Windows is free to move any nonlocked segments any time Windows is called. The result is that the this pointer could be invalid upon return from the call. Besides, Zortech only allows this to be far in Compact and Large model (you can't explicitly create a far object in Medium or Small model).

2) Each object has a pointer to a Virtual Method Table (VMT) which is usually in the DS of the program which created the object. This pointer is used to call the proper virtual function for an object. As long as you are calling from the same 'side' of the DLL-APP boundary which allocated the space for the object, the VMT can be found with a near pointer (eg calling ReDraw() for a cEllipse from within the APP). But, if you are calling a virtual function for an object that was allocated from the other side, you need a far pointer to find the VMT (eg calling ReDraw() for a cRectangle from the APP, or calling Paint() for a cEllipse from the DLL). Again ZTC only allows far VMT pointers when compiling with far data models (Compact and Large).

BC++ goes even further - VMT pointers are near in all models unless you 1) set a compiler option to put them in the code segment, or 2) declare the class as huge. Option 1) requires that code segments, like data, be locked and nondiscardable, while option 2) would not work properly with BC++ v2.0 (I couldn't derive a huge class from an _export base class).


1
0.5
0
-0.5



SOFTWARE

WE GUARANT
FIND A WIDE
SOFTWARE

"May I congratulate you on your continu
Software Paradise from all other ma

WE'LL BEAT ANY NATION

Software Paradise specialises in Mail-Order Software for IBM Personal Computers (and the Apple Macintosh) running DOS, Windows™, OS/2® and UNIX/XENIX®. Over 15,000 software products from over 2000 individual manufacturers guarantees you the most comprehensive range of PC software in the world from any single source. With up to 50% discount on all the latest titles from the U.S. and the U.K., Quality Service, Huge Stocks, Next Day Delivery, Pre & Post-Sales Technical Support and "We'll BEAT Any Nationally Advertised Price" - There's no need to call anyone else. Software Paradise (established in 1986) regularly supply nearly all of the U.K.'s leading corporate and public sector organisations, regional health authorities, universities, colleges and local government departments as well as thousands of small businesses and private users. Our policy of "Service First" and total commitment to our customers in the supply of quality products and advice has earned us a very large and loyal customer base. "*The Discerning Person's Guide to Software*" is the most complete Buyers Guide to PC software available with over 200 pages of applications and development tools. Call us on (0222) 887521 or complete the coupon to receive your FREE copy.

"The prize for the flashiest catalogue in my news-box, however, goes hands-down to Software Paradise .... the 200 pages are chock-a-block with interesting utilities, specialist libraries, wacky compilers and obscure operating systems." EXE, July '91.

FREEPHONE 0800 378·873

ORDERLINE ONLY

PLACE YOUR ORDERS FREE OF CHARGE
FROM ANYWHERE IN THE UK

ALL OTHER ENQUIRIES TEL: (0222) 887521 (15 LINES) OR FAX: (0222) 862209 (3 LINES)

Software Paradise, Avenue House, King Edward Avenue, Caerphilly, Mid Glamorgan, CF8 1HE

PARADISE

WE SUPPORT THE ETHICAL
USE OF SOFTWARE. TO REPORT
COPYRIGHT VIOLATIONS
CALL F.A.S.T. ON (0628) 660377
NEW! REACH US ON COMPUSERVE
AT MAILBOX 100015, 3570

EE YOU WON'T
ER RANGE OF
ANYWHERE†

d excellent service which distinguishes
order software suppliers." May '91

ALLY ADVERTISED PRICE*

What Makes Us Different

✓ FREEPHONE 0800 Orderlines
✓ FREEPOST Mailing Address
✓ 24-Hour FAX Orderlines
✓ Up to 50% Discount On Over 15,000 Software Titles
✓ Pre-Sales Consultation
✓ Post-Sales Technical Support
✓ Unbeatable Prices
✓ Next Day Delivery
✓ Huge Inventory
✓ Knowledgeable Sales Staff
✓ VISA, ACCESS & Mastercard Accepted
✓ No Credit Card Charges Until Shipping
✓ Corporate PO's Welcome
✓ Educational & Volume Discounts
✓ Latest Versions Guaranteed With Full Documentation & Manufacturers Warranties
✓ Over 6 Years Experience Providing The Very Best In U.K. and U.S. Software

Please Specify Media Size When Ordering
Goods Are Not Offered On A Trial Basis
All Prices Exclude Carriage & VAT
Prices & Specifications Subject To Change Without Notice

† The products you see listed in this advertisement represent a small sample of our full range

All Trademarks & Registered Trademarks Are Hereby Acknowledged

Software Paradise, Avenue House, King Edward Avenue,
Caerphilly, Mid Glamorgan, CF8 1HE

*** PRICE PROMISE**

If you find an identical software product advertised in the national computer press by another mail order company based in the U.K. at a legitimate, current price - which is less than our advertised price - we will happily BEAT that price and SAVE YOU MONEY! We will also beat other suppliers with our friendly and efficient service and very wide range of brand-name, quality software. Since 1986 we've served 1000's of customers with this same promise. Discover today why we're one of the fastest growing companies in the industry. To all of our current customers - Thank you for your support, we really do appreciate it!

Access
MasterCard
VISA

CIRCLE NO. 479

© Copyright 1991 Software Paradise. All Rights Reserved.

Mail to: Software Paradise, FREEPOST, Caerphilly CF8 1ZZ

- FREEPOST TODAY -

NO STAMP NEEDED

Please rush me my FREE copy of "The Discerning Person's Guide to Software"

NAME

COMPANY

ADDRESS

POSTCODE

TEL & FAX

EXE 2/92

PANEL® *Plus* II

Screen Generator and User Interface Library for 'C'

PANEL *Plus* II *features:* Interactive screen design editor • C and MS-FORTRAN code generators • library of user-interface functions • pop-up fields/windows • multi-line fields • mouse support • scrolling regions/ scroll bars • help boxes • pull-down menus • custom validation • shadow borders • complete **library source**, with low-level source and headers for all supported systems (DOS, Windows, OS/2, Unix, VMS) • interfaces to PC **graphics libraries** • OS/2 DLL • **no royalties.**

Also available: **Utility Source Option** which allows **PANEL *Plus* II** development on any suitable platform. *New in v2.2:* PANEL Plus II **Windows DLL**. Programs can be built for text mode or as Windows applications with **no source changes**. There is no need to redesign your PANEL Plus II application to run it under Windows.

Panel *Plus* II licence fee per programmer: **£285** (including library source and 700-page manual). With utility source option: £720. Please call for quantity pricing, full version list, and free demo disk.

Roundhill Computer Systems Limited

Orchard House, Ogbourne St. George, Marlborough, Wiltshire SN8 1SU
Telephone: **0672 84 535** • Fax: 0672 84 525

Roundhill also supplies development tools from Copia International, Graphic Software Systems, Lattice, Periscope, Pocket Soft and others. Please call for a full catalogue and price list.

CIRCLE NO. 480

Happily, if you run Windows in Standard (286 protected) or Enhanced (386 protected) mode, even 'locked' segments are moveable. After moving the data to a new segment, Windows just fixes up the selector table so that the same selector points to the new segment. Your program can use far pointers with wild abandon, while Windows efficiently scoots memory around.

Static Data Members

Remember that the DLL has a different DS from the APP. Static data of the base class is put in the DLL's DS. In order to access this data, you must be executing in the DLL (the DS register is set to point to the DLL's DS by the DLL function's prolog). Don't try to use static members in inline functions, as DS will be incorrect. Also, make sure that the user of your class can't do it either. The only way to guarantee this is to make all the static data private to the base class, and write functions to get and set the values. That's what I've done for `Piece` - all three static members are private, and I've defined functions for manipulating them (`SetContext()`, `GethDC()`). Remember, these functions can't be inline!

Member Functions

Functions that access only instance data can safely be inlined. `GetX()` and `GetY()` are examples of such functions.

Any member function that requires access to data in the DLL's DS should not be inlined, though. This includes functions which call the standard C library, or use any global data declared in the DLL.

Although you may think that your constructors and destructors only use instance data, they actually have secret implied calls to `new` and `delete`. Rather than calling `new` directly when you allocate an object dynamically, both ZTC and BC++ generate code which calls the constructor with a `NULL this` pointer. The constructor checks for this condition, and accordingly calls `new` if necessary. When a destructor is called, a flag is set if the data was dynamically allocated. If so, `delete` is called. If the constructor and destructor are defined in the DLL, the object will live in the DLL's heap, even though it was created by the APP.

Therefore, either all or (preferably) none of the constructors and destructors should be inline; they should all execute either in the DLL or in the APP to avoid situations where memory is allocated from one heap and released to another. `Piece` defines all constructors and destructors as non-inline.

Which Heap?

As mentioned above, every call to `new` from a constructor defined in the DLL gets memory from the DLL's heap. If several programs are using the DLL, it could exhaust the local heap (64 KB) quickly. To allow allocating more than 64 KB, you can redefine `new` and `delete` in the DLL, as I have at the top of PIECE.CPP. I simply defined `new` to allocate a fixed block of memory from Windows' Global Heap. Using the Global Heap is slower than the normal Local Heap, but you can allocate as much memory as you like, up to the limit of your RAM, or your swap file size.

I also redefined `new` and `delete` in the APP (PIECETST.CPP) to use the standard Windows Local Heap. The ZTC libraries do their own heap management in Compact and Large models under Windows, and I encountered bugs when running multiple instances of the same program using Zortech's heap management. Redefining `new` and `delete` solved the problem.

```
// #define   DLL   // to use as DLL

#include "Piece.h" // includes windows.h

#ifdef   DLL
int FAR PASCAL LibMain
      ( HANDLE hInst, WORD wDS,
        WORD wHeapSz, LPSTR lpszCL )
{ return 1; }

int FAR PASCAL WEP ( int bSystemExit )
{ return 1; }
//
#include  <stdlib.h> // to define size_t
// use Windows' Global Heap
void * operator new(size_t siz)
{
  return (void*)
  MAKELONG(0, GlobalAlloc(GPTR,siz));
}
void operator delete(void *obj)
{
  GlobalFree( (GLOBALHANDLE)
                LOWORD((LONG) obj) );
}

#endif
HWND Piece::hCurWnd = 0;
HDC Piece::hDC = 0;
PRECT Piece::Inv = 0;

void EXPORT Piece::SetContext
               (HWND hw, HDC h, PRECT i)
{  // set static data for Painting
  hCurWnd = hw;
  hDC = h;
  Inv = i;
}

// short, but *must not* be inline
HDC EXPORT Piece::GethDC()
{ return hDC; }

EXPORT Piece::Piece(HWND hw, int x,
                    int y, DWORD c)
{
  hWnd = hw;
  bnd.left = x; bnd.top = y;
  color = c;
  visible = 0;
}

// note:  Piece exported by DEF file
Piece::~Piece()
{ if (visible) Hide(DELAY); }

void EXPORT Piece::ReDraw()
{ // call paint if in Inv
  RECT r;
  if ( (hCurWnd == hWnd)  && visible &&
      IntersectRect(&r, &bnd, Inv) )
  {  // set brush, paint, reset brush
    HBRUSH hb, ohb;
    hb = CreateSolidBrush(color);
    ohb = SelectObject(hDC, hb);
    Paint(); // virtual, possibly in APP
    SelectObject(hDC, ohb);
    DeleteObject(hb);
  }
}

int EXPORT Piece::Contains(int x, int y)
{  // checks if (x,y) is inside Piece
  POINT p; p.x = x; p.y = y;
  return PtInRect(&bnd, p);
}

WORD EXPORT Piece::Hide(WORD when)
{  // hide this piece
  WORD was = visible;
  visible = 0;
  InvalidateRect(hWnd, &bnd, 1);
  if (when == IMMED)
    UpdateWindow(hWnd);
  return was;
}

WORD EXPORT Piece::Show(WORD when)
{  // unhide this piece
  WORD was = visible;
  visible = 1;
  InvalidateRect(hWnd, &bnd, 0);
  if (when == IMMED)
    UpdateWindow(hWnd);
  return was;
}

void EXPORT Piece::MoveAbs(int x, int y)
{     move upper left of piece to x,y
  WORD wasvisible = Hide(IMMED);
  bnd.right  = x+(bnd.right -bnd.left);
  bnd.bottom = y+(bnd.bottom-bnd.top);
  bnd.left   x; bnd.top = y;
  if (wasvisible) Show(DELAY);
}

// ** member functions of cRectangle

EXPORT cRectangle::cRectangle
         (HWND hw, int x,  int y,
          int dx, int dy, DWORD color)
  : Piece(hw, x, y, color)
{  // set rectangle to be drawn
  bnd.right  = x + dx;
  bnd.bottom = y + dy;
  Show(DELAY);
}

// declare destructor to ensure delete
// is called from proper context
// (inside DLL, not APP)
cRectangle::~cRectangle() {}

void EXPORT cRectangle::Paint()
{ // paint this Rectangle
  // color has been set by ReDraw()

  Rectangle(GethDC(), bnd.left, bnd.top,
                 bnd.right, bnd.bottom);
}
```

Figure 4 - PIECE.CPP

```
LIBRARY     PIECE
DESCRIPTION 'DLL for Piece class'
EXETYPE     WINDOWS
CODE        PRELOAD MOVEABLE DISCARDABLE

DATA        PRELOAD SINGLE FIXED
HEAPSIZE    1024
EXPORTS     __dt__5PieceFv
            __dt__10cRectangleFv
```

Figure 5 - PIECE.DEF

Remember, the location of object allocation is decided by where the code of the constructor and destructor reside, NOT where you call `new` and `delete`. Since `cRectangle`'s constructor and destructor are in the DLL, `cRectangles` are allocated on the Global Heap, while `cEllipses` (constructor and destructor in the APP) are allocated on the APP's Local Heap.

Virtual Functions and DS

Here's where it starts to get fun. Remember that when executing in the DLL, DS must be set to the DLL's DS, while in the APP it must be set to the APP's DS. As long as you don't cross the DLL-APP boundary, DS will be set properly, but any function that can be called from the other side (APP->DLL or DLL->APP) must have instructions to save the calling DS and point DS to the called procedure's heap. In terms of Figure 1, any crossing of the borders between CSes must be accompanied by an appropriate change to DS. Crossing from APP->DLL happens any time a base class member function is called. DLL->APP occurs when a function in the DLL calls a virtual function of a derived class (eg `Piece::ReDraw()` calls `cEllipse::Paint()`).

```
// PieceTst.cpp - test the Piece class
// and show how it should be used.
#include "piece.h" // includes windows.h

[illegible]
#include  <stdlib.h>  // defines size_t
   use Windows' Local Heap
void * operator new(size_t siz)
{
 return (void*) (void NEAR*)
   LocalAlloc(LPTR, siz);
}
void operator delete(void *obj)
{
  LocalFree( (LOCALHANDLE)
              LOWORD(obj) );
}
[illegible]                         ////////
   ** cEllipse - APP class from Piece

class cEllipse : public Piece
{
public:
  cEllipse(HWND hw, int x, int y,
           int dx, int dy, DWORD color);
   ~cEllipse();
  void Paint();
};

cEllipse::cEllipse(HWND hw, int x, int y,
              int dx, int dy, DWORD color)
: Piece(hw, x, y, color)
{  // set rectangle to be drawn
  bnd.right  = x + dx;
  bnd.bottom = y + dy;
  Show(DELAY);
}

cEllipse::~cEllipse()
{     // implicit call to Piece::~Piece()
}     // cEllipse::~cEllipse()

void cEllipse::Paint()
{     // call *only* from Piece::ReDraw()
  Ellipse(GethDC(), bnd.left, bnd.top,
                  bnd.right, bnd.bottom);
}
[illegible]

Piece *pieces[5]; // array of 5 Pieces
char szAppName[] = "PieceTest";

long FAR PASCAL  export
     WndProc(HWND, WORD, WORD, LONG);

int FAR PASCAL
WinMain(HANDLE hInst, HANDLE hPrevInst,
        LPSTR lpszCmdLine, int nCmdShow)
{
  HWND hWnd;
  MSG  msg;
  WNDCLASS wc;
  if (!hPrevInst)
  {
    wc.style  = CS HREDRAW | CS VREDRAW;
    wc.lpfnWndProc    = WndProc;
    wc.cbClsExtra     = 0;
    wc.cbWndExtra     = 0;
    wc.hInstance      = hInst;
    wc.hIcon          = LoadIcon(NULL,
                        IDI_APPLICATION);
    wc.hCursor        = LoadCursor(NULL,
                             IDC ARROW);
    wc.hbrBackground = GetStockObject
                        (WHITE BRUSH);
    wc.lpszMenuName  = NULL;
    wc.lpszClassName = szAppName;
    RegisterClass(&wc);
  }

  hWnd = CreateWindow(szAppName,
        "Testing Pieces",
        WS OVERLAPPEDWINDOW,
        CW_USEDEFAULT, CW_USEDEFAULT,
        CW_USEDEFAULT, CW_USEDEFAULT,
        NULL, NULL, hInst, NULL       );
  ShowWindow(hWnd, nCmdShow);
  UpdateWindow(hWnd);

  while (GetMessage(&msg, NULL, 0, 0))
  {
    TranslateMessage(&msg);
    DispatchMessage(&msg);
  }
  return msg.wParam;
}

long FAR PASCAL  export
WndProc (HWND hWnd, WORD msg,
         WORD wParam, LONG lParam)
{
  int ct;
  static curpc = -1, npc = 5;
  static xofs, yofs;
  switch (msg)
  {
   case WM CREATE:
     pieces[0] = new cRectangle(hWnd,
      20, 20, 150, 150, RGB(255,0,0));
     pieces[1] = new cRectangle(hWnd,
      40, 40,  80,  50, RGB(0,0,255));
     pieces[2] = new cRectangle(hWnd,
      70, 95, 120, 180, RGB(25,10,50));
     pieces[3] = new cEllipse(hWnd,
      200,100, 60, 90, RGB(255,255,0));
     pieces[4] = new cEllipse(hWnd,
      150,30, 120, 100,RGB(0,255,255));
     return 0;

   case WM PAINT:
     PAINTSTRUCT ps;
     HDC hDC = BeginPaint(hWnd, &ps);
     Piece::SetContext(hWnd, hDC,
                        &ps.rcPaint);
     for (int ct = 0; ct < npc; ct++)
       pieces[ct]->ReDraw();
     EndPaint(hWnd, &ps);
     return 0;

   case WM_MOUSEMOVE:
     if ((wParam & MK LBUTTON)
                     && (curpc != -1))
       pieces[curpc]->MoveAbs(
                   LOWORD(lParam)-xofs,
                   HIWORD(lParam)-yofs);
     return 0;

   case WM LBUTTONDOWN:
     curpc = -1;
     for (ct = npc-1; ct >= 0; ct--)
       if (pieces[ct]->Contains(
                        LOWORD(lParam),
                        HIWORD(lParam)))
       {
         curpc = ct;
         xofs = LOWORD(lParam)
              - pieces[ct]->GetX();
         yofs = HIWORD(lParam)
              - pieces[ct]->GetY();
         break;
       }
     return 0;

   case WM_RBUTTONDOWN:
     SendMessage(hWnd, WM LBUTTONDOWN,
                     wParam, lParam);
     if (curpc != -1)
     {
       npc--;
       delete pieces[curpc];
       for (ct = curpc; ct < npc; ct++)
         pieces[ct] = pieces[ct+1];
       curpc = -1;
     }
     return 0;

   case WM DESTROY:
     for (ct = 0; ct < npc; ct++)
       delete pieces[ct];
     PostQuitMessage(0);
     return 0;
  }  // switch (msg)
  return DefWindowProc(hWnd, msg,
                        wParam, lParam);
}
```

Figure 6 - PIECETST.CPP

DLL functions have code in the function prolog which saves the DS of the caller and sets DS to point to the DLL:

```
mov  AX, xxxx ;DLL's DS
...
push DS
mov  DS, AX
```

Before any statements of the DLL function are executed, DS is pointed to the DLL's heap. As there is only one instance of any DLL (multiple APPs use the same DLL instance), this method of loading a constant value works.

The same method won't work for an APP, though, as there can be many instances of the same APP, and each has it's own Local Heap (and DS). Therefore DS can't be loaded by a prolog to the function as was done in the DLL. Instead, each instance of an APP function which is exported (callable from outside the APP), is called through an 'Instance Thunk':

```
mov AX, xxxx ;this instance DS
jmp xxxx:yyyy;jump to function
```

The function begins by pushing DS, and setting DS to the value passed in AX. Each instance of an APP has it's own Instance Thunks.

The problem is that except for WindowProcs, Instance Thunks are not created automatically, they must be created by the programmer calling the Windows function `MakeProcInstance()`, which returns a pointer to the Instance Thunk, used thereafter to call the function. As the programmer doesn't have direct access to the Virtual Method Table (containing pointers to the virtual functions), this method of fixing DS is impractical for virtual functions.

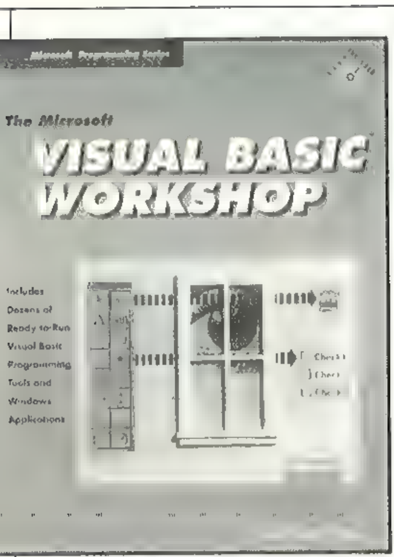
All is not lost, however, if you remember that the Stack Segment (SS) is always pointing to the current APP's stack, even when you are executing in the DLL. By definition, an APP's SS is equivalent to its DS. When a function in the APP (in particular, a virtual member function) is called from the DLL, its prolog can simply load DS from SS (after saving the calling DS, of course):

(Continued on page 66)

Microsoft Press

Essential books for Windows programmers

Microsoft PRESS®


Microsoft Programming Series
The Microsoft
VISUAL BASIC WORKSHOP
Includes Dozens of Ready-to-Run Visual Basic Programming Tools and Windows Applications

THE MICROSOFT VISUAL BASIC WORKSHOP

by John Clark Craig

The book that addresses the hottest issue in computing today - how to create full-featured Windows applications. This book-disk package explains Visual Basic concepts, techniques and tricks, and features a collection of 41 re-usable tools and application examples that can be incorporated into your Windows programming projects. The disk contains both source-code and executable files.

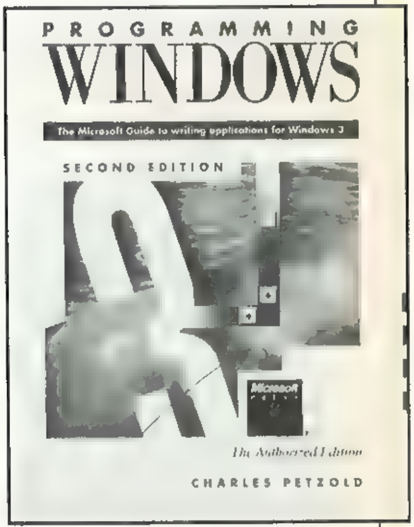
Paperback 350 pages £29.95 inc VAT
ISBN 1-55615-386-4

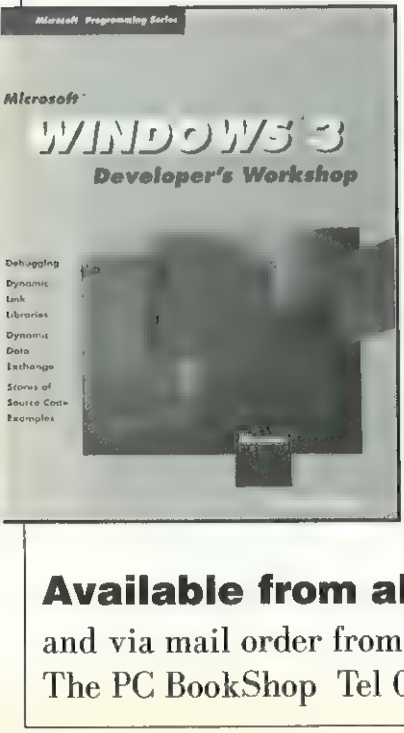
PROGRAMMING WINDOWS

by Charles Petzold

The best-selling, clear and authoritative book that no Windows programmer can afford to be without. It is packed with indispensable reference data, tested programming advice, and new sample programs. *"Just take it as a given - if you're going to program for Windows, buy this book. It will pay for itself in a matter of hours."* ***Computer Language***

Paperback 900 pages £27.95 ISBN 1-55615-264-7





MICROSOFT WINDOWS 3 DEVELOPER'S WORKSHOP

by Richard Wilton

"After you've nailed down the basics, Rick Wilton's valuable book will surely sharpen your Windows programming skills." ***Charles Petzold***

An example-packed resource, this book provides detailed discussions of some of the central and most complex issues faced by every Windows programmer. Topics include: Debugging, DLL, Custom Controls, Windows and Objects, DDE. In addition, Wilton offers practical approaches to programming puzzles.

Paperback 272 pages £22.95 ISBN 1-55615-244-2

Available from all good bookshops

CIRCLE NO. 481

and via mail order from:
The PC BookShop Tel 071 831 0022 Fax 071 831 0443

Visit our stand, No 139 at the Windows Show Olympia, 18-20th Feb

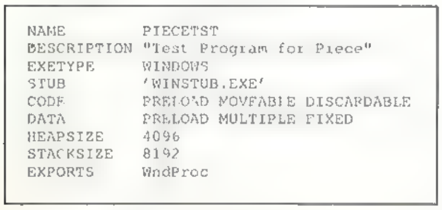
```
mov  AX, SS ;APP's DS
...
push DS
mov  DS, AX
```

In contrast to loading DS with a constant value (as done for DLLs), loading from SS will work with multiple instances of an APP.

Luckily, ZTC and BC++ both have a compiler switch that causes all far functions to load DS from SS as shown above. Use -W3 for ZTC, and -WS for BC++. As all far functions are considered exportable when using this switch, you don't have to mark the functions as `_export`.

Review

In brief, to put your class into a Windows DLL:



```
NAME        PIECETST
DESCRIPTION "Test Program for Piece"
EXETYPE     WINDOWS
STUB        'WINSTUB.EXE'
CODE        PRELOAD MOVEABLE DISCARDABLE
DATA        PRELOAD MULTIPLE FIXED
HEAPSIZE    4096
STACKSIZE   8192
EXPORTS     WndProc
```

Figure 7 - PIECETST.DEF

1) Use Large Memory Model - declare all Data Segments as FIXED.
2) Declare nonline members as `_export`.
3) Put destructors' mangled names in EXPORTS of the DLL's .DEF file.
4) Do not access static data or library functions from inline functions.
5) Make all static data members private to the DLL, and give access to them through (non-inline) member functions if necessary.
6) Do not make constructors or destructors inline.
7) Redefine `new` and `delete` in the DLL if you will need >64 KB of space for objects of classes defined in the DLL.
8) Compile the APP with -W3 (ZTC) or -WS (BC++) so that DS is loaded from SS on entry to all far functions.
9) Don't forget the fire extinguisher.

Conclusion

Although a price must be paid to put C++ classes into Windows DLLs, notably the requirement of using Large model and the necessity to be careful about static data and virtual functions, it is worth the expense. EXE files are smaller, and memory usage is potentially less (if several APPs use the same DLL).

Due to data segments being locked, programs which use these DLLs will run in a degraded state in Real mode Windows, but will show little performance penalty in Standard and Enhanced mode.

EXE

Laine Stump has been threatening to get into Windows programming for at least two years, but has just managed to get off his duff in the last six months. His desk is now several inches deep in purple and blue manuals, dirty dishes, and half-eaten cheese sandwiches.

As usual, he can be reached via the PC Tech BBS at (0101-612-345-4656, evenings, US time) or by post at: Bilkent University, Lojmanlari 3/9, 06533 Bilkent / Ankara, TURKEY.

The code accompanying this article is available on disk. Send a blank formatted floppy disk to the Editor, following the instructions given on Page 1, column 1. Mark your envelopes 'CPPDLL'.

FORTRAN

"**FTN77/386** is simply the best FORTRAN compiler on *any* platform, *anywhere* - and believe me, I've tried them all." (available for 386 or 486, DOS or Unix)

"We bought **INTERACTER** as a graphics library for PC , Unix and VAX, but now we use it for menus, input screens and system access too. INTERACTER makes our code user-friendly ***and*** portable."

"We started with 150,000 lines of vintage FORTRAN... **SPAG** translated it to beautifully structured Fortran 77 which worked first time. The whole operation took 2 man days; we estimated 6 man *months* without SPAG. ***That's*** what ***I*** call productivity"

STOP PRESS The *plus***FORT** toolkit (which includes SPAG) has been greatly enhanced. New features include a global static analyser, standardisation of declarations, automatic clutter removal, variable renaming and many more.

Call today for a free demo diskette featuring these (and other) products.

If you use FORTRAN, you MUST check this one out.

Polyhedron Software Ltd.

Polyhedron Software

Linden House, 93 High Str. Tel. (44) 0865-300579
Standlake, Witney OXON Fax. (44) 0865-300232
OX8 7RH United Kingdom Compuserve 100013,461

US TOLL-FREE FAX 1-800-777-5519

CIRCLE NO. 483

INNOVATIVE DESIGN TOOLS

FROM GREAT WESTERN INSTRUMENTS

Great Western Instruments are the specialists in innovative design tools for the embedded systems developer. Our services include full technical and design consultancy support for all our products.

NEW PRODUCTS

XMEM - Virtual memory manager using base, expanded, extended or disk Similar to malloc. Ideal for real mode memory hungry applications. With source code. **£145** **Circle 418**

MEMORY TOOLKIT - Library of routines to manage expanded (EMS) and extended (XMS) memory. Big DOS arrays can be created in EMS or XMS. With source. **£95** **Circle 419**

EMPTY SHELL- C routines for freeing up memory when shelling to a large program. Also saves program screens and graphics mode. Uses expanded, extended memory or disk to save swapped code. **£165** **Circle 420**

OVER-C+ - Compact non-preemptive scheduler with message queues, triggers, timers and memory management. Supports Borland C++ and MS C. Examples included for embedded and PC applics. **£345** **Circle 421**

AMX 86/386/68000/Z80 - Real time executives from KADAK. Fast, reliable, compact and ROMable. Preemptive task scheduler with optional time slicing. PC and embedded applications using C, C++ and assembler. From **£995** **Circle 422**

EMBEDDED DOS - Fully MS-DOS 3.31 compatible. Optimised for real time with multitasking, reentrant. For PC and embedded targets. Very low royalties. **£455** **Circle 423**

SELECT CASE - World renowned lowcost, PC based CASE tools supporting SSADM, Yourdon / Hatley / Ward Mellor or HOOD. Easy to use. Single user **£495.** Project-SELECT for LAN **£995** **Circle 424**

SMS - Fully featured software management system. Easy to use. Version Control, Configuration, Change and Modification Request Management. Available for MSDOS, OS/2, OS-9, UNIX and VMS. From **£490** for single user MSDOS. **Circle 425**

The above is only a fraction of the full range of design tools available.

GREAT WESTERN
-Understanding the needs of other designers!

For immediate information about these & other products in our range, use your Fax/Telephone to call **Fax-direct** (0865) 727232, **Item 444401**

GREAT WESTERN INSTRUMENTS

Unit 1m Farrington Fields
Farrington Gurney
Bristol BS18 5UU
Tel 0761 452116
Fax 0761 453226

VAT Extra Carriage Free

All trademarks acknowledged

CIRCLE NO. 484

Desktop FORTRAN

EXPERTS AGREE WHAT TO BUY – THEY JUST CAN'T AGREE WHY!

"The FTN77 compiler has allowed us to port our main-frame applications to 386 micros with minimum effort. The speed of compilation is incredible!. . . The run time performance is fantastic, full support for 32-bit operations and the 80387 coprocessor means our programs run like lightning!"
Tony Fitzpatrick, Exploration Consultants Ltd

"I find the FTN77/386 compiler to be fast and easy to use . . . . We often download programs under development from a mainframe to the PC to run through this computer."
Rob McLaren
FTN77/386 User

"Without the FTN77/386 compiler it would have been totally impractical for us to convert our million-line + power system analysis program (PSS/E) to run on a 386 PC."
Wayne B'Rells
Power Technologies Inc

"FTN77/386 provides more intrinsic functions than any other' functions to manipulate strings handle screen and keyboard I/O, read the date and time, parse the command line Advanced file manipulation functions allow an FTN77/386 program to work with files in a C-like manner"
Scott Ladd,
Computer Language, Nov 1988

I am very impressed and have decided to shift all our production programs over to FTN77. . ."
Dr G.R Chaplam
NCL Investments Ltd

"The ease of use, speed, debugging facilities, dynamic libraries with resulting small .EXE files, and extensive library routines made it a joy to use."
H. Gordon Jensen
FTN77/386 User.

"By far the fastest and most complete FORTRAN compiler that I have ever seen or used in the DOS environment".
Tim Wool
FTN77/386 User.

"The debugger was very easy to use. There is an on-line help facility which made reference to the manual unnecessary."
. . Mike Gunn and Arul Britto,
EXE Magazine, May 1989

"For sheer speed and sleekness the (Salford) compiler stands out like a Ferrari among Yugos"
Jack W Crenshaw, Computer Language, May 1990

FAST, FRIENDLY & FULLY FEATURED *- THE PERFECT MIX - CALL US NOW!*

SALFORD SOFTWARE LTD • VENABLES BUILDINGS • 5 COCKCROFT ROAD • SALFORD M5 4NT
TEL: 061 745 5678 • FAX: 061 745 5666

FULL WINDOWS 3.0 SUPPORT!

FTN77 386/486 – extending MS-DOS.

UNIX VERSION AVAILABLE TOO!

CIRCLE NO. 485

Gyre and Gimble

...are Julian Bucknall's words for the action of Turbo Pascal's overlay manager, in the first of a new column devoted to !C (= 'not C', geddit?) programming.

There are two alternative beginnings to this article: choose the one you believe (a bit like *The French Lieutenant's Woman* in reverse).

The first was when I was recently composing an internal memo on how Turbo Pascal V6.0 used memory to overlay my application's code. I blithely wrote something like '...and the code is switched in from EMS in chunks of 2 KB to 5 KB...', when I stopped. How did I know how big my overlaid units were? With Turbo Pascal's smart linker stripping out unused procedures, functions and methods the code size reported by TPC (the command line compiler) was no help. The MAP file gave me information I didn't believe (the overlaid units looked miles too small); the manual is silent on the subject.

The second story was a program crash. My error handler from hell gave me the address of the failing statement: bang in the middle of my overlay buffer. Yeah, well, thanks and everything, but which routine in which overlaid unit was crashing at that location? I mention in passing that the application has about 100 overlays, and more Turbo Vision spaghetti than Italy.

So what was I to do? I needed more information on the overlay system. I picked up the Turbo Pascal manual, dusted off Turbo Debugger, and descended into virtual 8086 mode.

Basics

Overlays are a method of fitting a very large program into a restricted area of memory. Part of the program is compiled into a separate overlay file, and a clever overlay manager pulls in code blocks from this file into a common buffer when they are needed. A code block which is no longer being used can be discarded in favour of another code block. The overlay system in Turbo Pascal V6.0 is very easy to implement (a couple of switch settings and some code preamble to set up the manager), and even has some rudimentary tuning parameters to help it run more smoothly.

I shall be describing the overlay system as implemented in Turbo Pascal V5.5 and V6.0 (they are the same); I no longer have V5.0 and so am unable to check whether the following is substantially true or not. I would recommend at this point that you read the Turbo Pascal manuals on the overlay system (for V5.5 see the *Reference Guide* and the *OOP Guide*, for V6.0 see the *Programmer's Guide*); I shall assume from now on that you have a good knowledge of the overlay system as described officially by Borland. I'm afraid I also assume some rudimentary assembly language knowledge as well.

If you're feeling adventurous, you could investigate how to move the overlay buffer

In at the shallow end

First point of attack is all those undocumented identifiers, described oh-so-tersely in the manuals. These are presented in Table 1, together with any extra information I've gleaned on the official ones. Before looking at it, I must introduce a definition. You all know what the segment part of an address refers to (the offset in paragraphs from the start of memory). Well, the overlay manager uses another type of segment value, one that is measured from the start of the program code itself. Every time one of these segment values is used, it has to be translated (relocated) to the absolute segment form. This is easy: just add the Program Segment Prefix segment (`PrefixSeg`) and the size of the PSP in paragraphs ($10). I shall call this unrelocated

```
type
  PUnitStubBlock = ^TUnitStubBlock;
  TUnitStubBlock = record
    usbInt3F     : word;
      { Always $3FCD, the INT 3Fh instruction }
    usbRetOfs    : word;
      { Offset of first return address }
    usbFileOfs   : longint;
      { Offset of unit in overlay file }
    usbCodeSize  : word;
      { Unit's code size in bytes }
    usbFixupSize : word;
      { Size of fix-up data block in bytes }
    usbEntries   : word;
      { Number of entries in the jump table }
    usbNextBlock : word;
      { Locale of next stub block }
    usbBufferSeg : word;
      { Segment of unit in overlay buffer }
    usbWasCalled : word;
      { 1 if unit was called on probation }
    usbNextLoad  : word;
      { Locale of next loaded stub block }
    usbFiller    : array [1..5] of word;  { Unused (?) }
  end;
```

Figure 1 - The unit stub block

System Science

• Software • Specialists • Software • Specialist • Software • Specialists • Specialist • Software • Specialists •

Unix

SCO Unix Op. Sys **SCO** Unix Dev Sys.
SCO Open Desktop **SCO** ODT Dev Sys
SCO TCP/IP & NFS **FTP** PC/TCP
Interactive Unix Op Sy / Architect Series
Informix 4GL & SQL **RM** Cobol-85
LPI Compilers **C Scape** Screen Lib
LPI C++ (native) 386 Unix & Sun- **NEW**
... many more for 386 Unix and Sun

Mathematics

Derive - The Mathematical Assistant
Mathematica 2 Windows (now in), 386. MAC
Mathematica for Workstations (Sun, HP..)
MathCad for Windows (new) GrafTool
What's Best! **Lindo**
SPSS/PC, Statgraphics Chiwriter Sci WP

Windows Development

(see Microsoft and Borland, Zortech)

CASE:W Corp.	£645.00
Smalltalk/V Windows	£295.00
Btrieve for Windows	£345.00
Object Trieve	£call
Commonview	£395.00
Win++ by Blaise	£150.00
Windows Control Pallette - Blaise	£95.00

CrossDevelopment

2500AD Cross Assemblers	£135.00
2500AD Simulator/Disassemblers	£115.00
Cross Compilers - Introl, Manx, Hitech	

Pascal (see Borland)

TopSpeed Pascal SE	£115.00
Object Professional	£110.00
Btree Filer MU	£120.00
Asynch Professional	£89.00
Blaise Turbo Vision Toolkit	£89.00
Blaise Turbo Asynch Plus	£115.00

Power Basic	£89.00
many Basic libraries for comms, graphics	
PC Logo	£50.00
LMI UR/Forth	£295.00
Smalltalk V/DOS	£69.00
muLISP 90	£225.00
RM Cobol-85 (new Dev Pack)	£call.
MS DOS 5.0, QEMM 386 etc	£call.

Microsoft

Quick C for Win **NEW!**	£99.00
Microsoft C 6.0	£220.00
MS Fortran 5.1 (new ver)	£210.00
MS Windows SDK	£210.00
MS Win SDK & C6.0	£335.00
MS Basic Compiler 7.1	£225.00
MS Cobol 4.5 (DOS & OS/2)	£460.00
MS Macro Assembler 6.0	£79.00
MS Visual Basic	£99.00
MS Quick C 2.5	£60.00
MS Quick C & Assembler	£99.00
MS Quick Basic	£60.00
MS Source Profiler (new)	£49.00

Authorised Languages Dealer

Borland

Borland C++ & AFW **new 3.0**	£295.00
Borland C++ (Dos / Win) **3.0**	£195.00
Turbo C++ Windows **NEW!**	£85.00
Turbo Pascal Windows	£99.00
Turbo C++ & Turbo Vision **NEW!**	£75.00
Turbo Pascal 6	£75.00
Turbo Pascal 6 Prof	£135.00
Turbo Debugger and Tools	£89.00

Authorised Languages Dealer

Nu-Mega Tech.

Bounds Checker	£169.00
Soft-Ice/W (Windows)	£259.00
Soft-Ice (for 386 DOS)	£259.00
CV/1 (CodeView in a Window)	£135.00

Intersolv *(Authorised)*

PVCS for Dos	£375.00
PVCS Config Build (PolyMake)	£165.00
PVCS Networks, OS/2 & Sun	£call
PVCS Professional Editor (SPE)	£189.00

MKS

MKS Toolkit new ver 3.2	£155.00
MKS RCS new ver 5.1	£155.00
MKS Lex & Yacc new ver 2.6	£155.00
MKS multi-user and OS/2	£call

Tools and Editors

Brief 3.1	£195.00
Kedit (Xedit for PC)	£99.00
.RTPatch for DOS, Win & OS/2	£245.00
.RTLink Plus	£295.00
Plink 86 Plus	£275.00
Personal Rexx new ver 3	£110.00
Ghost (auto software testing)	£135.00
EasyCase Plus	£345.00
C Programmers Toolbox	£175.00

... and many more tools and utilities

C++ Compilers

see also Borland

Zortech C++ Dev Ed (incl OS/2)	£345.00
Zortech C++ Sci & Eng	£525.00
JPI TopSpeed C++PExtend	£175.00
JPI TopSpeed C++ SE	£115.00
JPI TopSpeed C++ PE	£175.00
JPI TopSpeed C++ Pro Dev.	£229.00

C Compilers

see also Microsoft

Watcom C 8.5	£275.00
Watcom C386 8.5 with ADS Supp	£495.00
Aztec C 86 (with ROM support)	£265.00
TopSpeed C PE	£175.00

C Datafile

CodeBase 4.5 for C, C++, Win- new	£235.00
Btrieve - DOS or Windows ver	£345.00
Ctree Plus from Faircom	£345.00
Faicom Professional Toolbox	£745.00

C Communications

Essential Comms	£175.00
Greenleaf CommLib	£195.00
C Asynch Manager (Blaise)	£115.00

C Screens

CScape with Look & Feel	£365.00
Vermont Views 3	£335.00
Panel Plus II	£275.00
Zinc for C++ (Win &Dos)	£225.00

C Graphics

Essential Graphics & Chart	£195.00
PCX Programmers Toolkit	£165.00
Halo	£195.00
Metawindows	£195.00

... many more libraries

Pharlap 386 DOS Ext SDK	£325.00
Pharlap 286 DOS Ext SDK (new)	£325.00

Fortran Compilers

Lahey F77L	£375.00
Lahey F77L-EM/32 with OS386	£875.00
Watcom F77 386 8.5 with ADS supp	£495.00
Salford FTN 77-386	£765.00
PC/Interacter (screens)	£325.00
Ingraf Graphics source	£195.00

... many Fortran Libraries

- Prices are exclusive of VAT.
- Prices include delivery to GB.
- Prices are subject to change.
- VISA, Access and Mastercard welcome with telephone orders.

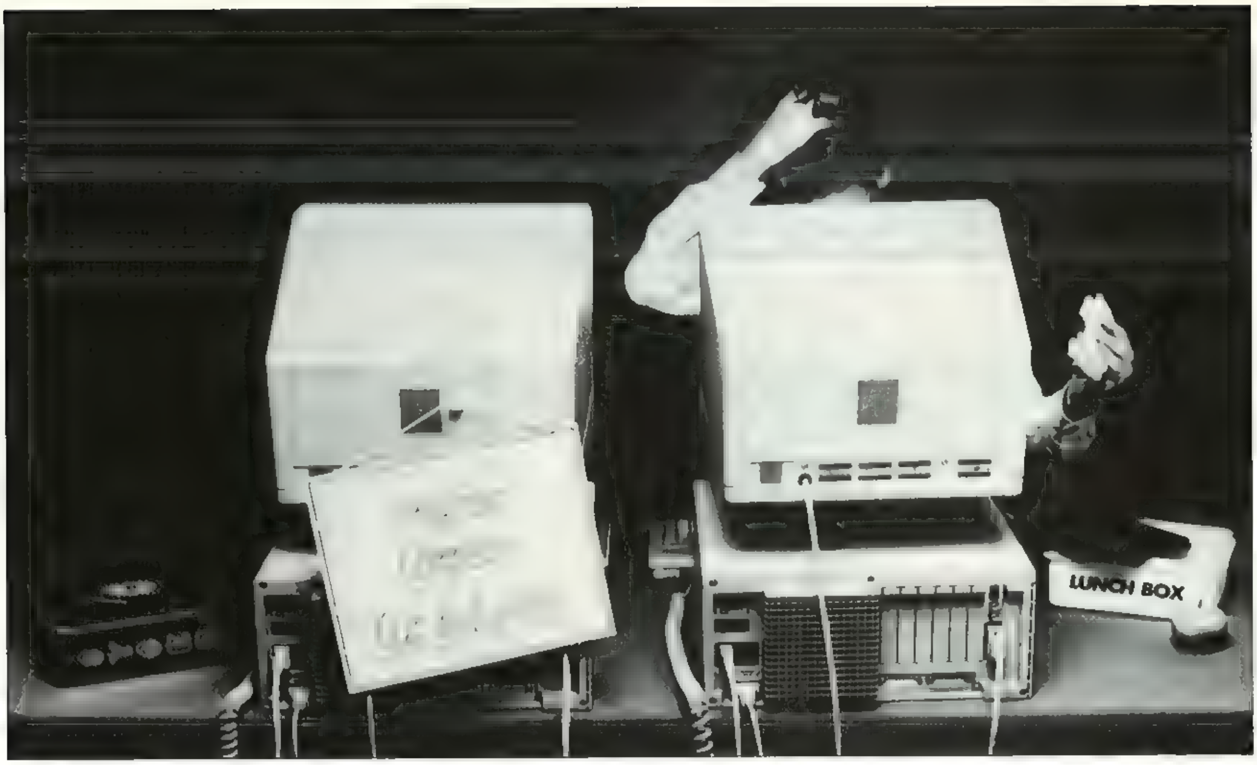
3-5 Cynthia St
London N1-9JF
Fax: 071 837 6411

071 833 1022

• Software • Specialists • Software • Specialist • Software • Specialists •

System Science

CIRCLE NO. 486


LUNCH BOX

segment a *locale*; this conjures up a mental image of the words local and address so it shouldn't be too difficult to remember (if it catches on, you read it here first!).

As you can see from Table 1, we have some intriguing variables (actually typed constants) to play with later on. First, however, a basic question. How **does** the overlay manager know when an overlaid routine gets called? Enter the unit stub block and the jump table.

When your program gets linked, each of your handcrafted overlaid units gets put into the .OVR file, and a 'stub' is left in the .EXE file itself, one for each unit. This 'stub' comprises the unit stub block (my term, I haven't a clue what it's called at Borland) and its jump table. The unit stub block is 32 bytes long and has the layout shown in Figure 1. Immediately following this stub block is a 'jump' table. Each element of this table is 5 bytes long, and there is one entry for each exported routine from your overlaid unit. A call to a routine in an overlaid unit will actually call the relevant jump table entry. The overlay manager 'notices' (exactly how is explained later!), and if the code for the routine is not yet in the overlay buffer, the unit is loaded, and the jump table adjusted to point to the code. Similarly, if the unit gets moved in the overlay buffer, the jump table gets adjusted again. When the unit gets overwritten ('removed' if you will), the jump table is adjusted once more.

`OvrReadBuf` (Procedure variable)

Allows you to intercept overlay load operations by writing your own read routine. Documented a bit in the manual. Could be used, for example, to write a routine to load overlays that have been compressed, or that reside in XMS. Although it is documented which overlay is required (parameter `OvrSeg` is just the segment of the unit stub block, described in the text), where it goes and whether any fix-up exercise has to be done is totally undocumented (the answers are at `ubsBufferSeg`, and yes!). `OvrInitEMS` replaces the standard loader with its own.

`OvrCodeList` (Word)

The manual calls this the 'Overlay code segment list'. This is a 'locale' (defined in the text), and it points to the first unit stub block. The unit stub blocks form a chain which you can 'walk' to get information about any overlay unit.

`OvrLoadList` (Word)

The 'Loaded overlays list'. Again a locale, but this points to the first overlay stub block whose code is currently loaded somewhere in the overlay buffer. These stub blocks form a chain which you can 'walk' to get information about any unit that is currently loaded. In other words, it provides the map of the overlay buffer. The unit code block pointed to by this variable is the 'oldest' unit in the overlay buffer and will be the first to be removed, ie it's the tail of the ring buffer.

`OvrDebugPtr` (Pointer)

The 'Overlay debugger hook'. Used by the overlay manager to tell Turbo Debugger that it has just completed a probation trap or unit load operation. This should be a far procedure with no parameters. ES is the unit stub block segment, BX the offset of the called routine in the unit code block.

`OvrHeapSize` (Word)

The 'Initial overlay buffer size'. True, but it's quoted in paragraphs not bytes (cf `OvrGetBuf`).

`OvrHeapPtr` (Word)

The 'Overlay buffer pointer'. This segment value actually points to the bottom of the free space in the overlay buffer. The next overlay unit will be loaded here. The head of the overlay buffer.

Table 1 - Undocumented Identifiers

Going deeper

Slightly more detail now. Each jump entry is initially set by the compiler to an INT 3Fh instruction (2 bytes) followed by the routine's offset in the unit (2 bytes) followed by a zero byte. Suppose an overlaid routine gets called and there is nothing in the overlay buffer. As mentioned before, the call goes to the start of the jump table entry for the routine. The INT 3Fh gets executed and lo! the overlay manager gets control (`OvrInit` installs it as the INT 3Fh interrupt handler). At this point the stack looks like Figure 2.

Excellent! The segment word of the return address is none other than the segment of the unit stub block, the offset word points directly to the routine's offset in the unit's code block. The overlay manager can now work out the location of the unit in the overlay file (field `usbFileLoc`). The unit's code is loaded (together with the fixup data block) into the overlay buffer at `OvrHeapPtr`, any fixups are done (and the fixup data block discarded), and the jump table entries are all altered to 5-byte far JMPs (we now know the segment address of the code in the buffer, and the offset was in the jump table to begin with). `OvrHeapPtr` is altered to point at the next segment in the overlay buffer. By manipulating the stack, the overlay manager ends by transferring control to the newly loaded routine. From now on, calls to this routine will meet with a far JMP straight to the code, and the overlay manager will neither know about nor interrupt them.

Easy-peasy, huh? However, it gets more complicated when units are about to get discarded from the buffer. Let us ignore probation for the moment and assume unit A has been in the overlay buffer for a while. Another unit needs to be loaded and the overlay manager notices that unit A will have to go (it's at the tail of the buffer). Now some of A's routines may be active, ie they have made calls to other routines and are waiting for these routines to return. The return addresses are on the stack. If we remove the unit, these return addresses will be pointing in the middle of some other code in the overlay buffer. The overlay manager needs to fix the stack so that it gets control when these returns are executed. To do this, it has to assume two things: (1) all RETs are far (ie all routines run up to now have been compiled with the far model: the return addresses will have segment and offset components), and (2) all routines push BP (the stack frame pointer for their caller) as their first action. Turbo Pascal does this automatically for you when you declare procedures and functions as far; beware, however, if you do some assembler programming.

The overlay manager can now 'walk' through the stack until it finds a return address segment for the unit about to be overwritten. It saves the corresponding return offset, and replaces the full return ad-

Top of stack segment
...
...
...
Original Caller's return segment
Original Caller's return offset
Flags
Return segment from the INT 3Fh
Return offset from the INT 3Fh

Figure 2 - The stack the overlay manager sees

```
procedure OvrTracker; far;
  { OvrTracker gets called whenever the overlay  }
  { manager is about to relinquish control.      }
  { Install with the statement:                  }
  {   OvrDebugPtr := @OvrTracker;                }

  const
    LoadCount : word = 0;  { Tracks OvrLoadCount }

  var
    StubBlock  : PUnitStubBlock;
    RoutineOfs : word;

  begin
    { When called, ES is the segment of the unit }
    { stub block, BX is the offset of the routine}
    { in the unit itself.                        }
    asm
      xor AX, AX
      mov StubBlock.Word[0], AX
      mov StubBlock.Word[2], ES
      mov RoutineOfs, BX
    end;

    { Increment the number of times StubBlock    }
    { has been trapped.                          }
    { The 5th filler word is used as a counter.  }
    inc(StubBlock^.usbFiller[5]);

    { If OvrLoadCount has changed, the unit used }
    { by StubBlock has just been loaded.         }
    { The 4th filler word is used as a load      }
    {   counter.                                 }
    if (OvrLoadCount <> LoadCount) then
      begin
        LoadCount := OvrLoadCount;
        inc(StubBlock^.usbFiller[4]);
      end;

    { At this point, you may call another        }
    { UNOVERLAID routine, and pass StubBlock and }
    { RoutineOfs as parameters if needed.        }

  end;
```

Figure 3 - The OvrTracker procedure

dress with the unit stub block segment, zero offset (ie the INT 3Fh instruction in the stub block: ah ha, light dawns!). The saved offset is placed in field `usbRetOfs`. The manager continues its walk, and all other return addresses for the same unit will have only their segment word changed, to point at the unit stub block (don't worry, they are never executed in this half-and-half state). When the program eventually executes a RET to a removed routine, the unit stub block will get called instead. The INT 3Fh is executed, and the overlay manager gets control. It notices that it was called due to a RET (the last word on the stack will be a 2 - think about it), loads the unit, and patiently refixes all the unit's return addresses on the stack (ie all return addresses that have a segment equal to the unit stub block). Finally, it pushes the return address that should have been used (the segment is the unit's new segment in the buffer, the offset was stored at `ubsRetOfs`) onto the stack, and issues a RETF.

By now you should recognise that the stack fixing stuff gets done every single time a unit gets loaded, or indeed every single time it gets shuffled around the buffer. If the unit is in the buffer, its return address segments in the stack point into the buffer, if not, they point to the unit stub block. In fact, every time the overlay manager gets called, it walks the stack and fixes the segments for any overlaid unit it finds. Better safe than sorry.

The probation scheme works in roughly the same way. When the overlay manager notices that a unit has entered the probation area, its jump table is altered to the INT 3Fh version. If the unit gets called on probation, the `usbWasCalled` flag gets set and the jump table altered to far JMPs once more. When the unit reaches the tail of the buffer, the overlay manager notices the flag has been set, resets it, moves the unit's code from the tail to the head of the buffer and gives the unit another ride around.

Time to Interfere

The procedure in Figure 3 (OvrTracker) is a debug routine whose address you can install into `OvrDebugPtr`. OvrTracker uses two unused fields at the end of each unit stub block to track probation and load calls. As mentioned in Table 1, the debug procedure must be far, with no parameters. On entry, ES is the segment of the unit stub block in question (BX is significant as well, it is the offset of the routine to be called). We don't know explicitly (as far as I know) whether the procedure was called due to a probation trap or a unit load, just that it will get called as the last thing the overlay manger does before returning to the main program. I've declared a single typed constant (mimicking `OvrLoadCount`) to track the load calls. Install `OvrTracker` immediately you've initialised the overlay manager, the manager's EMS driver (if used), and have finished altering the overlay buffer size.

The code is small and easily understandable. One warning, do *not* put this procedure in an overlaid unit: the program will nose-dive at Mach 2. At the end of the procedure is a comment showing where you could place a call to another procedure, for example to dump the overlay buffer map to disk or printer. Again, to be pedantic, do *not* put this called procedure in an overlaid unit, and do not allow it to use any overlaid routines in its turn. Remember when this procedure is called, the overlay manager is still active and it definitely is not re-entrant.

```
function RealAddr(Locale : word) : PUnitStubBlock; near;
  { RealAddr calculates the actual address of a  }
  { PUnitStubBlock from a locale value           |
  begin
    { A locale of zero = the end of the chain    }
    if (Locale = 0) then
         RealAddr := nil
    else RealAddr := Ptr(Locale + PrefixSeg + $10, 0);
  end;

procedure OvrWalker(FullList : boolean);
  { OvrWalker walks through the full unit stub   }
  { block list if FullList is true, otherwise it }
  { walks through the list of loaded stub blocks.}
  var
    StubBlock  : PUnitStubBlock;

  begin
    { Get the first stub block in the chain }
    if FullList then
         StubBlock := RealAddr(OvrCodeList)
    else StubBlock := RealAddr(OvrLoadList);

    { Walk through the chain ... }
    while (StubBlock <> nil) do
      with StubBlock^ do
        begin
          { Get next stub block in the chain }
          if FullList then
               StubBlock :  RealAddr(usbNextBlock)
          else StubBlock := RealAddr(usbNextLoad);
        end;
  end;
```

Figure 4 - The OvrWalker procedure

Ask about our NEW Windows 3 Version

A poor workman will always blame his tools . . .

. . . But what if the tools are less than the best?

A craftsman is only as good as his tools and a good craftsman always uses the best tools for the job. ED provides a rich, powerful text editing environment with all the features you need for programming in the 90s.

Ease of use, congfigurability and attention to detail are paramount in ED's design. You will be up and running right out of the box. ED even emulates other editors, Brief, Wordstar, Norton editor and Qedit . . .

ED's capabilities leave all other text editors behind and coming from QBS you get a level of service and support nobody else provides.

If you currently use cumbersome, slow, old fashioned tools, then it's time for a change. Take control, move into the fast lane with ED today, you owe it to yourself . . . . **ONLY £99**

Call for a free DEMONSTRATION DISC

Some of ED's Highlights

- Edit files to 100MBytes+, with lines up to 1024 characters.
- Use screen resolutions of 132 x 60 or more.
- Multiple windows and files.
- Cut and Paste text between files and windows with just a few keystrokes.
- Full support for rectangular Column blocks.
- Unlimited ***UNDO*** and ***REDO***.
- Easy to use pull down menus, context sensitive help, help bars and quick reference guides.
- Full ***LAN*** and multiuser support.
- Run compilers with error tracking.
- Releases all memory for your compiler.
- Language sensitive editing for ***C/C++***, ***dBase***, ***Clipper***, ***DataFlex***, ***PFX+***, ***Pascal*** etc.
- Fast ***C extension language*** for ultimate control. Includes source code for nearly all editor commands.
- Regular expression search and replace.
- Smart indenting, templates, object matching, code completion etc.
- Named keyboard macros.
- Flexible, fast, easy to use and fully configurable.
- Calculator, Calendar, Line drawing, ASCII chart etc.
- Save/restore editor state between sessions, including files and windows.
- Clipboard, Text buffers and Bookmarks.
- Run any DOS program or Shell to DOS.
- Flexible printer support and background printing.
- Directory Tree.
- Word wrap and Paragraph reformat.

Plus much more.

Soft as it Gets

Ed
the programmer's editor

© Soft As It Gets 1991

QBS Software Limited,

10 Barley Mow Passage,
London W4 4PH

Tel: 081-994 4842
Fax: 081-994 3441
BBS: 081-747 1979

"Thanks for a great product. It's just what I've been looking for in an editor. I have tried many editors in my time, including Wordstar, Brief, Norton Editor, VIPC but none match ED's power and flexibility."
David Bagnara, Bell Organisation Pty Ltd, Victoria

"I have deleted Brief from my machine." *Damian Lewis, Kenneth A. Hansen & Associates P/L, Victoria*

"Brilliant editing system."
Geoffrey Evelyn, Tyne & Wear

CIRCLE NO. 488

```
UNIT OvrUsage;

{$O-} { No overlays allowed here }

INTERFACE

uses Overlay;

IMPLEMENTATION

type
  PUnitStubBlock = ^TUnitStubBlock;
  TUnitStubBlock = record
    usbInt3F      : word;
    usbRetOfs     : word;
    usbFileOfs    : longint;
    usbCodeSize   : word;
    usbFixupSize  : word;
    usbEntries    : word;
    usbNextBlock  : word;
    usbBufferSeg  : word;
    usbWasCalled  : word;
    usbNextLoad   : word;
    usbFiller     : array [1..3] of word;
    usbOurLoads   : word; { 4th filler - redefined }
    usbOurTraps   : word; { 5th filler - redefined }
  end;

{ The saved previous exit procedure }
var
  ExitSave : pointer;

{ Procedure OvrTracker is the essentially the }
{ same as that in Figure 2; we don't want the }
{ routine offset this time however.           }
procedure OvrTracker; far;
  const
    LoadCount : word = 0;
  var
    StubBlock  : PUnitStubBlock;
  begin
    asm
      xor AX, AX
      mov StubBlock.Word[0], AX
      mov StubBlock.Word[2], ES
    end;
    inc(StubBlock^.usbOurTraps);
    if (OvrLoadCount <> LoadCount) then
      begin
        LoadCount := OvrLoadCount;
        inc(StubBlock^.usbOurLoads);
      end;
  end;

{ RealAddr calculates the actual address of a }
{ PUnitStubBlock from a locale value          }
function RealAddr(Locale : word) : PUnitStubBlock; near;
  begin
    { A locale of zero signifies the end of a chain }
    if (Locale = 0) then
         RealAddr := nil
    else RealAddr := Ptr(Locale + PrefixSeg + $10, 0);
  end;

{ Procedure ReportUsage is designed as an exit }
{ procedure. It reads the MAP file to get the  }
{ unit names, and then dumps the unit usage    }
{ statistics to file OVERLAY.DMP.              }
{ WARNING - contains NO error checking.        }
procedure ReportUsage; far;
  const
    MaxUnits = 100; { The max no of units we cater for }
  type
    UnitName = record { To store data for each unit }
      Locale : word;
      Name   : string[8];
    end;
  var
    i, ec        : integer;
    StubBlock    : PUnitStubBlock;
    StubLocale   : word;
    F            : text;
    MapFileName  : string;
    Line         : string absolute MapFileName;
    NameArray    : array [1..MaxUnits] of UnitName;
    Finished     : boolean;
  begin
    ExitProc := ExitSave;

    { Assume we're running under MSDOS 3.0+, and }
    { the MAP file name is the same as the EXE   }
    { file name with a '.MAP' extension.         }
    MapFileName := ParamStr(0);
    MapFileName[0] := char(Pos('.', MapFileName) - 1);
    MapFileName := MapFileName + '.OVR';

    { Open the MAP file }
    Assign(F, MapFileName);
    Reset(F);

    { Miss the first three lines (the headings) }
    for i := 1 to 3 do readln(F, Line);

    { Read through the MAP file until we get to  }
    { the the line for the heap. For each line   }
    { read, get the start locale and unit name   }
    { and store in the local array NameArray.    }
    i := 0; Finished := false;
    repeat
      readln(F, Line);
      inc(i);
      with NameArray[i] do
        begin
          Line[1] := '$';
          Val(Copy(Line, 1, 5), Locale, ec);
          Name := Copy(Line, 23, 8);
          if (Name = 'HEAP    ') then Finished := true;
        end;
    until Finished;
    Close(F);

    { Open the statistics file OVERLAY.DMP, }
    { output the headings.                  }
    Assign(F, 'OVERLAY.DMP');
    Rewrite(F);
    writeln(F, 'Overlay usage for ', ParamStr(0));
    writeln(F);
    writeln(F, 'Name----  -Size  Traps  Loads');

    { Now walk the unit stub block chain }
    StubBlock := RealAddr(OvrCodeList);
    StubLocale := OvrCodeList;
    while (StubBlock <> nil) do
      with StubBlock^ do
        begin
          { Find the unit name }
          i := 0;
          while (NameArray[i].Locale <> StubLocale)
          do inc(i);

          { Write details to the Usage file }
          writeln(F, NameArray[i].Name,
                     usbCodeSize:7,
                     usbOurTraps:7,
                     usbOurLoads:7);

          { Get next stub block in the chain }
          StubBlock := RealAddr(usbNextBlock);
          StubLocale := usbNextBlock;
        end;
    Close(F);
  end;

var
  OvrFileName : string;

begin
  { Assume we're running under MSDOS 3.0+, and   }
  { the overlay file name is the same as the EXE }
  { file name with a '.OVR' extension.           }
  OvrFileName := ParamStr(0);
  OvrFileName[0] := char(Pos('.', OvrFileName) - 1);
  OvrFileName := OvrFileName + '.OVR';

  { Initialise the overlay manager, with a buffer}
  { twice the size of the original. Set the      }
  { probation area.                              }
  OvrInit(OvrFileName);
  OvrSetBuf(OvrGetBuf * 2);
  OvrSetRetry(OvrGetBuf div 3);

  { Install OvrTracker. }
  OvrDebugPtr := @OvrTracker;

  { Install the ReportUsage routine as an exit }
  { procedure.                                 }
  ExitSave := ExitProc;
  ExitProc := @ReportUsage;
end.
```

Figure 5 - An example Overlay Usage unit

5 out of 5 hackers prefer other software protection methods to Hardlock E-Y-E.®

What hackers dislike...
Hardlock E-Y-E was designed using cryptographic principles. It took the experience and know-how of Germany's No. 1 in software protection and the leading edge technology of a US semiconductor company to create the ultimate software protection tool. Hardlock E-Y-E is based on a custom chip featuring secure algorithmic response rather than simple bit swapping or counting schemes.

What software developers like...
Hardlock E-Y-E combines all the features software developers require in a single product: algorithmic response to provide security and an optional non-volatile memory to allow custom configurations. FAST Electronic has made implementation of Hardlock E-Y-E in your software easy. Use HL-Crypt to protect .EXE or .COM files, or incorporate high level language interface routines in your software. The algorithm parameters and the contents of the memory can be programmed in seconds using our Crypto-Programmer card. This unique card guarantees that no one else can burn your original codes. Simply plug the card into any PC slot and start up your own Hardlock E-Y-E workshop.

What your customers will like...
Hardlock E-Y-E allows unlimited backup copies. The device is shipped with the software for the user simply to plug into the parallel interface and forget. Daisy chainability, outstanding reliability (no battery is needed), and the most compact High-Tech design ensure that your customer will accept Hardlock E-Y-E.

What your accountant will like...
Hardlock E-Y-E needs no factory coding. This ensures optimum delivery schedules and stock flexibility. Revenues will go up as software piracy and multiple usage are prevented. Despite its wealth of features, Hardlock E-Y-E's prices remain competitive.

...As more and more software developers, customers and accountants appreciate the Hardlock E-Y-E device, hackers like it less and less.

Hardlock E-Y-E
programmable, algorithmic response and memory option – all in one.

MAGNIFEYE

Order your demo unit today. Contact Magnifeye,
235-239 Walmer Road, Walmer Studio #6, W11 4EY, Telephone 071 221 8024, Fax 071 792 3449.

CIRCLE NO. 489

The procedure in Figure 4 (OvrWalker) enables you to walk the unit stub block chains: both the full chain and the loaded units chain (the overlay buffer map), its action dependent on a passed boolean parameter. The former mode could be used in an exit procedure to dump the usage of the overlaid units at the end of the program (especially if you've installed `OvrTracker` as well). The latter mode could be called by `OvrTracker` itself, to dump the overlay buffer map to disk every time the overlay manager gets control. As an alternative, you could install it in your own error handler from hell, to be called when a run-time error occurs in the overlay buffer.

The only problem with `OvrWalker` and the unit stub blocks is that you've no real idea which unit they're talking about. Enter the MAP file. If you compile your program with the TPC compiler and use switch /GS, you'll create a MAP file showing all the segments in your program: code, data, stack, and heap. Each line in this MAP file shows the start and end segment values (locales in my parlance), the size in bytes, the name and class of each segment. If you walk through the unit stub blocks with `OvrWalker`, their locales can be matched to the start locales in the MAP file. The best thing to do would be to simply dump the unit stub blocks to a file from within your program, and then write an analysis program to operate on this file. By reading and parsing the MAP file first, the analysis program could easily attach meaningful unit names to each of the stub blocks. In Figure 5, I show the code for a unit (with no error checking for brevity's sake) that initialises the overlay manager, installs `OvrTracker`, and, on program termination, uses the MAP file to produce a simple usage dump of the overlaid units.

Conclusion

As I have shown, the Turbo Pascal overlay manager is extremely sophisticated, both from an external viewpoint and also from watching its internal machinations.

From the undocumented features I have laid bare, you can now find out how best to use overlaid units, which ones to overlay, and conversely, which ones not to. You can find out by observing your own program whether the recommended probation size (one third of the overlay buffer) is the best for you. You can decide whether to split up large overlaid units or not.

If you're feeling adventurous, you could investigate how to move the overlay buffer, how to shrink it, and (harder) how to grow it dynamically, as your program relaxes and intensifies its demands on the heap. Perhaps, like me, you could write your own unit to load overlaid units from XMS rather than EMS, disable the 64 KB EMS page frame, and reuse the memory for something else (the overlay buffer?).

EXE

Julian Bucknall has been designing and programming for 12 years, and full-time in Turbo Pascal for the last three. His other favourite language used to be RPG III, so there's no hope. Off work, he can be found at the wheel of his red Volvo 1800S in and around London, a Saintly vision.
Turbo Pascal V6.0 Professional is a product from Borland International (0734 321150), and is available from all the dealers, eg Grey Matter (0364 53499) asks £137.

SOLUTION SYSTEMS PROGRAMMING TOOLS GIVE YOU LEVERAGE.

U.K. VERSIONS • DUAL MEDIA •

Put the physics of programming to work for you. ¶ Right out of the box, together or separately, Solution Systems products give you leverage that will send your productivity soaring. It all begins with BRIEF. Our programmer's editor is the favorite of programmers around the world, renowned for its speed, configurability, and ease of use. ¶ dBRIEF for X-Base and Paradox are award-winning add-ons that custom tailor BRIEF for DBMS development. BRIEFor C++ adds a C++ browser to BRIEF for those of you with an object orientation. ¶ Sourcerer's Apprentice provides version control that's seamlessly integrated with BRIEF. It's never been easier to keep track of development on individual PCs and Novell or UNIX-served networks. ¶ And then there's C-Worthy. The User Interface Development System. C-Worthy gives you the menus, screens, colors, and screen designer you need to create great-looking applications at warp speed. ¶ When you're ready to wrap it all up, CHARGE, an interactive code profiler with direct access to BRIEF, will give you new insight into your code's performance. ¶ Raise your productivity and creativity to new heights. Apply the physics of programming to your applications today.

AUTHORISED DEALERS:-
Absolute Computing 0923 56043 • **Bonsai** 071-631 5454 • **Computer Solutions** 0932 352744 • **Computacentre** 0923 56040 • **Computerland** 0256 816611 • **Corporate Computers** 0245 450039 • **Grey Matter** 0364 53499 • **International Software** 081-479 0049 • **Metrocom** 081-740 6666 • **P&P Corporate** 0706 217744 • **Roundhill** 0672 84535 • **Software Box** 0904 783036 • **Software Construction Co.** 0763 244114 • **Software Corporation** 0904 691391 • **System Science** 071-833 1022 • **Technomatic** 081-205 9558 • **Technology Plc** 0925 818220 • **Worldwide Computers** 081-543 2211.

Solutionsystems

1 The Maltings, Green Drift,
Royston, Hertfordshire SG8 5DB
Telephone: 0763 244141
Facsimile: 0763 244025

CIRCLE NO. 491

Towards a fully object-oriented Clipper

Nantucket is evolving Clipper away from its dBASE compiler origins towards an OOP future, as Chris Sennitt explains.

Nantucket has been talking about an object-oriented Clipper since 1988. This article looks at how far down the road Nantucket are with this technology; which problems have been tackled by the current (5.01) release of Clipper, and those which remain to be addressed.

When considering Clipper's evolution, it's important to understand the enormity of the problem that Nantucket faces: how to convert the thousands of Clipper programmers to OOP without frightening them into the arms of, say, Fox or Borland. Many Clipper programmers come from a non-technical background, having started out using a 'faster dBASE'. The strategy Nantucket has adopted appears to be the evolutionary approach, with incremental upgrades gently easing the Clipper programmer into a new ways of solving complex business problems.[1]

This strategy has worked well with earlier upgrades. User Defined Functions, for example, were a simple extension to the old procedure mechanism: change the word procedure to function, add a return expression and you have a function not a procedure. Although OOP is slightly more complex, the current campaign to upgrade the current user base from Summer '87 version to the 'OOPier' 5.01 release appears to be working.

```
local aFreq[10], aHtz

// Copies the array pointer aFreq to aHtz

aHtz := aFreq

aFreq[1] := 1066

? aHtz[1]              // Outputs 1066
```

Figure 1 - Assigning Clipper array variables

Under the surface

What was in the current release of Clipper 5.01 reflecting the modernisation of the language? There was a pre-processor and a handful of operators borrowed from C, and a sophisticated Virtual Memory Management system which allowed programs to handle up to 16 MB of variable storage.

But most notably *missing* from this release was a full implementation of OOP. Instead, Nantucket supplied four fixed classes, a Table Browser (`TBrowse`), a Table Column (`TBColumn`), an Error class (`Error`) and a Get class (`Get`). There was no facility to define or create new classes. These four fixed classes have had a mixed reaction. For the novice they are confusing, for the adventurous Clipper programmer it's just enough to whet the appetite, and for the full professional it's not enough.

Nantucket has recently announced that the next release of Clipper 5 will have a full implementation of OOP, with an object-oriented SQL Interface to follow. Although Nantucket was 12 months late with Clipper 5, I think the steps taken towards OOP so far indicate that the full OOP version is not too far off.

What steps has Nantucket made towards OOP? For a start there's the business of multi-typed variables. Clipper is a dynamically-typed language; so you can write

```
mixup := 1
```

and subsequently code

```
mixup := "Surprise!"
```

and go on to assign a date to `mixup`, then a logical, then an array etc. While this feature is ripe for abuse by inexperienced programmers, it does have legitimate uses. Clipper Summer '87 was limited in this area; arrays could not be passed to and from functions, and array elements could not contain other arrays. Release 5.01 removed these limitations. Arrays are now no different to any other variable, and elements of arrays may now hold any data type, including objects and arrays. This has OOP implications because, from the point of view of the Clipper language, objects and arrays are very similar. An object's instance variable can hold any data type, including arrays and objects. So we can imagine an object containing an instance variable that contains an array of objects that each contain an instance variable that has an array... to any depth.

Then there's Clipper's system of garbage collection. The Virtual Memory Management system, combined with its dynamic typing, allows Clipper to recover unused memory automatically. In most typed, declarative languages the scope and lifetime of a variable (and its memory space) can be calculated at compile time and code is generated to reclaim this memory. In some languages, destruct messages may be sent that clear down other run-time resources such as file handles. With a non-declarative, dynamically typed language, scoping rules and the lifetime of some variables can only be deduced at run-time.

To understand the problem, look at the implementation of arrays. It's important to distinguish between the space allocated to a Clipper variable and the separate piece of memory allocated to its data. Internally the variable contains a reference to the array memory, in much the same way that a C `char *` points to the actual string. The pointer stuff is completely hidden from the Clipper programmer (thus removing the requirement for 90% of the debugging ses-

sions performed by his C-bound brother!), although it becomes obvious from the way arrays are used (See Figure 1). We now have two variables referring to the same array, the rule in Clipper is that an array does not get garbage-collected as long as something live points to it. Clipper's VM system is very sophisticated; it does not use reference counting to decide if something can be reached as this is too easily fooled (see Figure 2). Clipper incrementally scans memory variables tagging blocks of memory that can be reached. Memory that is not tagged is released for future use.

In C++ an object may be compared to a C structure plus methods associated with that structure. Internally, 5.01 objects are arrays with associated methods. So the VM usage for objects is identical to arrays: they live as long as they can be reached.

There's also the *send* operator. Implementation of the Clipper language is split into two parts, the compiler and the run-time system. The 5.01 Compiler now understands the send operator ':' (eg `object:message`). This generates very simplistic code that 'sends' the object to a run-time system function, along with the message and any parameters. The run-time system looks up the method associated with the message for that class and passes control to this method along with the parameters. So both the compiler and the run-time system have some understanding of OOP. All that is missing from the standard Clipper is the ability to define your own classes.

So the internals of 5.01 are already object-oriented; this is how the supplied four fixed classes can operate. Third party products, like my own company's *SuperClass*, use these internals to implement a full object-oriented extension to 5.01 and have been available since May 1991. SuperClass allows the application programmer to define new classes via some User Defined Commands. These commands load the internals of Clipper with the definition of the classes, number of instance variables and the associated messages and methods. The fixed classes supplied with 5.01 load the internals in the same way, so the add-ons are just exploiting the built-in support for objects.

A Good Question

'If all this is already in place, why hasn't Nantucket implemented full OOP?', you cry. I believe Nantucket could have implemented an OOP extension between the releases of 5.0 and 5.01 (a six month period). Here at ChyDale Software, it took us a matter of weeks to get SuperClass into alpha-testing, and that's without access to the run-time source code.

```
// The fooling of the reference counter

do while .t.           // loop forever
   foolit()
enddo

function foolit()
   local a[10], b[10]  // both arrays have a reference count of 1

   a[1] := b           // increase reference count of b by 1
   b[1] := a            / increase reference count of a by 1

return( nil )

// return will remove variable a and b thus reducing the
// reference count of the array referred to by a and b by 1
// Final count would be 1 so they stay in memory but cannot
// be reached.

// Clipper does not use this Primitive mechanism
```

Figure 2 - Problems with reference counters

I believe that Nantucket is delaying for two reasons: as a continuation of its gently, gently approach to introducing New Things; and to enable it to keep Clipper in step with its future technology (ie the Aspen 'Clipper-for-Windows' project). By moving forward at 'just the right speed', it hopes to take a significant number of Clipper programmers into OOP. By developing the next DOS version of Clipper, code-named '5.X', in tandem with Aspen, Nantucket hopes to create a class definition syntax which is sufficiently robust to be used in both systems. This will help in the transition from Clipper 5.X to Aspen.

Possible Features

Now we have looked at the steps taken towards a full OOP Clipper, it is reasonably simple to deduce many of its features. The syntax is unlikely to be far from that of SuperClass and the other add-ons available. There are only so many ways to describe the names of the instance variables and the names of the methods, and in the past, Nantucket has not been afraid to borrow ideas from other languages.

That said, I expect other features to be added. For example, many SuperClass users have asked for mechanisms to control the assigning and accessing of instance variables. Suppose we have an object `Obj` with a piece of instance data `Dat`. Then the action

```
Obj:dat := 4
```

would cause a special piece of code, defined by the application programmer, to be called. C++ programmers will recognise this as analogous to overloading the = operator. Access to variables could be similarly regulated (there is no direct analogy for this in C++).

Most of the current add-ons allow the programmer to create private, public and read-only instance variables. I feel that this is not the Nantucket way. Access and assign routines could easily replace the need for these mechanisms, or alternatively the SuperClass mechanisms could easily be simulated with access and assign mechanisms.

Let's look at the possible syntax for defining an object. In this late-binding, dynamically-typed language, not much is required to define an object: a list of its instance variables and the code for the methods. Figure 3 shows a typical definition for a SuperClass object. Some lines may not be required by a fully OOPed Clipper that understands class definitions completely. These lines are indicated by comments. Figure 4 shows the syntax for single inheritance and the overriding of an inherited method (`init`). Although SuperClass supports multiple inheritance, Nantucket has made it clear that this feature will not be in the next release of Clipper.

Many of the internal features of 5.01 suggest that a lot of the implementation ideas have been derived directly from the Smalltalk language, most notably code blocks, arrays and objects. However, I doubt that Clipper will support for class objects. These are used in Smalltalk (and other late binding languages) to hold information about the class, with each class defined by exactly one class object. In Clipper, it's possible to hold equivalent information about classes by using class variables, which can easily be implemented using static variables in the class definition file.

Destruct

One of the problems that remains for Nantucket to solve is that of the destructor mechanism found in C++. Because C++ is declarative and uses early binding, it's possible to predict when an object 'goes out of scope', and it can then be arranged for a destruct mechanism to be called. In Clipper, the problem is much harder. An object 'dies' only when it can no longer be reached

MACHINES OF EXCELLENCE

386SX 16Mhz £799.00. Super VGA Monitor, 1Mb Ram, 40Mb Hard Disc Enhanced keyboard

KJD computer systems are manufactured in the UK using the finest quality components. Our machines are rigorourly tested and configured to ensure both reliability and performance

NEW MACHINES

MACHINE	Mono Herc	Mono VGA	Colour VGA	S. VGA
286 12Mhz	£475	£535	£660	£710
286 16Mhz	£500	£560	£685	£735
386 SX-16	£600	£660	£785	£799
386 SX-20	£650	£710	£835	£885
386 DX-25	£745	£805	£940	£990
386 DX-33 CACHE	£899	£950	£1070	£1120
486 SX-20 CACHE	£1045	£1115	£1280	£1320
486-25 CACHE	£1345	£1415	£1580	£1630
486-33 CACHE	£1600	£1660	£1785	£1835

All the above are supplied with 1.44Mb drive, 40Mb Fast IDE hard disk, Enhanced Keyboard, Serial & Parallel ports, Flip-top case, 1Mb RAM.

WE WILL MATCH YOUR BEST PRICE AND GIVE YOU £80 WORTH OF SOFTWARE FREE!!

- ★ All Machines Use Top Quality Components
- ★ 1 Year Back-To-Base Warranty
- ★ All Machines Are Totally Upgradeable
- ★ Big Quantity Discounts Given
- ★ Trade Enquiries Welcome
- ★ Part Exchange Welcome
- ★ Data Transfer Available

Please add £55 for DOS. All other options available to your specification. Please Telephone for Prices.

Upgrade Your IBM XT Clone To An AT 286!

Make your machine run up to 11 times faster!

By allowing us to exchange your old eight slot XT clone motherboard with a new AT 286 compatible board running at 11 times the speed of an average XT That's almost twice the speed of many AT's!

We will fit a special 286 board with eight bit expansion slots to allow the use of your old eight bit XT cards. (This includes hard disk controller cards which can cause problems with normal 286 boards)

The XT to 286 upgrade fitted with 512K RAM is available for just

£165 + VAT & Carriage

PHONE FOR DIY KIT!!

This upgrade is also available with 640K or 1Mb RAM

Or even Upgrade your 286 to 386!

We can upgrade a standard 286 machine with 1Mb RAM to a 16Mhz 386SX with 1Mb for just **£275** + Carriage & VAT

NOTE - AMSTRAD USERS

SPECIAL DEALS ARE AVAILABLE TO UPGRADE YOUR MACHINE TO 286, 386SX OR 386 DX! SEE EXAMPLE PRICES BELOW!

Part-Exchange Your Amstrad PC2086. Hard Disk, Machine For The Following:
All Machines with 16 Bit & 8 Bit Slots. 1Mb RAM, Fast 40Mb Hard Disk. (Data Transferred Free Of Charge). 102 Key Keyboard. Attractive Baby AT Flip-Top Case. VGA Card (For Use With Existing Amstrad VGA Monitor).

12Mhz 286 (Specification As Above)	**£385**+VAT
16Mhz 386SX (Specification As Above)	**£525**+VAT
25Mhz 386DX (Specification As Above)	**£675**+VAT

SVGA COL. MONITOR, TILT & SWIVEL STAND (1024 x 765 x 256)	£225
MONO HERCULES MONITOR, TILT & SWIVEL STAND	£65
MOUSE (MS OR PC COMPATIBLE), 100, 200, 400, 800 DPI, MAT, 25 9 WAY CONVERTER, MOUSE HOLDER & SOFTWARE	£25
MS DOS 5	£65
MS DOS 4	£60
MS DOS 3.3	£50
FLIP-TOP CASE & PSU, 3x 5.25" + 1 x 3.5" BAYS	£60
ENHANCED KEYBOARD XT/AT	£30
40Mb I.D.E. HARD DISK	£135
5.25" 1.2Mb DRIVE	£49

3.5" 1.44Mb DRIVE	£42
3.5" FLOPPY DRIVE FRAME (MATCHING)	£7
MULTI I/O, FLOPPY/IDE HARD DISK CONTROLLER, 2S, 1P	£24
MONO GRAPHICS CARD/LPT1 PRINTER PORT	£14
VGA CARD 8/16 BIT 256K	£45
SUPER VGA CARD 256K	£65
SUPER VGA CARD 512K	£72
SUPER VGA CARD 1Mb	£82
MIRACLE MENU V2	£40
MIRACLE LABELS V1.5	£40
MIRACLE INVOICES V1	£40

NEXT DAY DELIVERY **NOTE: All Prices Exclude VAT & Carriage**

All prices and specifications subject to availability and change without notice

KJD COMPUTER SYSTEMS Phone or Fax 0424 721761

Unit 8-10 Mercatoria Business Centre 100-102 Norman Road St Leonards-On-Sea East Sussex TN38 0EB

CIRCLE NO. 494

NEW UNBEATABLE PRICES ON OUR

"The Best low cost alternative personal computer"
P.C. World

Atomstyle 386 SX came top in group review.
P.C. Plus June '91

On the APC1 486/33, "You can't get a system much faster than that, and it's just the job for a programmer with a slow compiler".
P.C. Plus

WINTER COLLECTION

286 16MHz	386SX 20/25 MHz	386 25MHz	386 33MHz	386 40MHz	486 33MHz
(SY 228) Expandable to 4MB on board.0 wait state. Landmark speed 21MHz. Expansion slots: 6x16 bit, 1x8 bit.	*(SY 366)* Expandable to 8MB on board.0 wait state. Landmark speed 26MHz. Expansion slots: 6x16 bit. Call for Slim Case option.	*(SY 334)* Expandable to 8MB on board.0 wait state. Landmark speed 33MHz. Expansion slots: 6x16 bit, 2x8 bit.	*(SY 374)* 64K CACHE MEMORY Expandable to 32MB on board + EMS. Landmark speed 58.7MHz. Expansion slots: 7x16 bit, 1x8 bit.	*(SY 380)* 64K CACHE MEMORY Expandable to 32MB on board. 0 wait state. Landmark speed 70MHz. Expansion slots: 6x16 bit, 2x8 bit.	*(SY 433)* 64K CACHE MEMORY Expandable to 32MB on board. 0 wait state. Landmark speed 156MHz. Expansion slots 6x16 bit, 2x8 bit.
£535	£595 £4.22 PER WEEK	£695 £4.92 PER WEEK	£775 £5.43 PER WEEK	£810 £5.66 PER WEEK	£1095 £7.04 PER WEEK

SYSTEM SPECIFICATION *1MB Memory (4MB on 486) * 40MB Hard Disk * 14" MVGA Monitor 800 x 600 * 3½" 1.44MB Floppy * 102 Keys Keyboard * 1 Parallel/2 Serial * Desk Top Case * Built by highly trained technicians * All systems easily upgraded and all systems can be configured to customers specifications.

FREE
The following are supplied FREE with each system purchased
*1 YEAR on-site maintenance
(Subject to registration card return)
* DOS/MOUSE
* DUST COVERS

OPTIONAL EXTRAS Upgrade your 286/386/486 with these ***SUPER VALUE add-ons.***
(Please add to price of system).

SCREEN TYPE

Hi Res VGA 800 x 600 VGA Card 256K	£90
Super VGA Res. 1024x768 256 colours VGA Card 1MB	£140
Non Interlaced	£190

HARD DISK

105MB IDE 18m/s	£100
180MB IDE 18m/s	£195
330MB SCSI 17m/s	£595
520MB SCSI 16m/s	[illegible]
1.2GB SCSI 13.5m/s	£1169
IDE Cache Controller 1-4MB	£215
SOFTWARE MS DOS 5 0	£25

MEMORY 1MB (extra)	£35
FLOPPY DRIVE 5¼" 2MB	£40
CASES Mini Tower	£20
Tower	£55

APC 7

* Fax/Scanner
* Printer * Copier

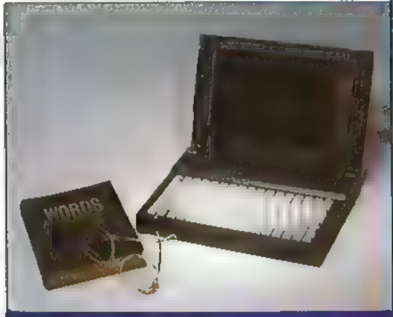
The ideal low cost machine for the small business

NEW — £425

NOTEBOOK COMPUTER

*** 386SX 20MHz * 60MB HD**

* 1.44MB Floppy
* **2MB** Memory
* VGA Mono 640 x 480
* 32 Gray scales
* RS232/Parallel/ Mouse/Scanner
* Rechargeable/ removable NiCAD battery 2.5hrs.



NEW PRICE £1195

NETWORK SERVICE

In-House Network Technical Expert
NOVELL
RED BOX LTD AUTHORISED DEALER

NEW SERVICE

Fully Tax Deductable LEASING for companies

START UP PACK

Installation plus ONE HOUR TUITION by highly qualified computer technician **£35**

LONDON
TEL: 081 801 1838
FAX: 081 365 1110
Atomstyle Limited, Unit 15,
Lockwood Ind.Park, Millmead Road,
Tottenham, London N17 9QU

LEEDS
TEL: 0532 442767
FAX: 0532 451980
Atomstyle Limited
Unit 21a Baker House,
9 New York Road, Leeds LS2 9PF

Super UltraSlim VGA Work Station
40 Megabyte only £695

All registered Trade Marks acknowledged.

ESTABLISHED 1982
HARD TO BEAT FOR PRICE & SERVICE!

APC ATOMSTYLE

All sales subject to Atomstyle Limited Terms and Conditions. Prices subject to adjustment without prior notice. All goods subject to availability

ALL PRICES EXCLUDE VAT AND CARRIAGE. ORDERS UNDER £50 SUBJECT TO £10 SURCHARGE. WE WELCOME ENQUIRIES FROM DEALERS, EXPORT, OFFICIAL COMPANY GOVT EDUCATIONAL ESTABLISHMENTS.

CIRCLE NO. 495

YES Send this coupon back FREEPOST

Please call □
Please send me data on the following
286/386/486 Systems □ Notebook computer □
Network Services □ Workstations □ APC7 □
Laser/Matrix Printers □ Peripherals □
Please use BLOCK CAPITALS
Name____________________
Company____________________
Address____________________
____
Post Code__________Tel__________
Please add my name to your mailing list □
.EXE 2/92

by the VM system. We then hit a *Catch 22* problem: how do you send a destructor message to something you can't reach? If you can send it a message it must be reachable, and if it's reachable then you should not send it the destructor message. What is wanted is the ability to call the destructor as soon as the object is no longer reachable, and in reverse order of creation. It is not obvious how this can be achieved.

Preparing for the Future

If all this OOP is being introduced with half an eye on Aspen, what is Aspen all about? Mainly speed and Microsoft Windows. Aspen will include some very sophisticated run-time/compile technology that should enable a ten-fold increase in performance over Clipper 5.X. It will probably include optional type declarations that will give 'hints' to the compiler and run-time system for method look-up and instance variable access/assign. This should then give the best of both worlds, a late binding language that can be optionally early bound.

Many Clipper programmers are trying to write systems today that will easily transfer to both Clipper 5.X and Aspen. It is certainly Nantucket's goal to convert as many of

```
#include "SuperCls.ch"        // remove this line
#include "Inkey.ch"

local nKey, oImage


oImage := saveScreen():Init(  4, 4, 22, 47 )
do while (nKey:= inkey()) !=  K ESC     // The escape key 27
   do case
   case nKey == K_DOWN         // Down Arrow
      oImage:down()            // Move the image down one line
   case nKey == K_UP           // Up Arrow
      oImage:up()              // Move the image up one line
   endcase
enddo

Return

Class saveScreen

   var  top, right, bottom, left, image

// Remove from here


   Message init( t, r, b, l)
   Message up()
   Message down()
   Message display()

end class

// to here

Method init(t, l, b, r)
   self:Image := savescreen( t, l, b, r)
return( self )

method up()
   self:top --
   self:bottom --
return( self )

method down()
   self:t ++
   self:b ++
return( self )

method display()
   restscreen( ::top, ::left, ::bottom, ::right, ::image )
return( self )


// Note :: is a short hand notation for self:
// implemented with a #define :: self:
```

Figure 3 - Defining a Clipper object

these programmers as possible. So what do you have to do today to achieve this?

You should certainly avoid older style Summer '87 functions and commands. These are clearly marked in the manual and in the on-line guides. Declare all variables before they are used (LOCAL or STATIC) - this is a declaration of the variable's name, **not** its type. Avoid PRIVATE and PUBLIC variables, the scoping and lifetime rules make them very difficult to compile and optimise. Start to convert the systems over to OOP as soon as possible. Always compile with a /W warning flag, as this will notify you of any variables that are used without being declared.

End thoughts

So what have Nantucket got that's new to the industry? Very little, actually. Most of the features have been derived from other languages (code blocks, pre-processor, new operators, late binding, dynamic typing, non-declarative, virtual memory management). What is new is that all this is built around the Clipper language, something that's easily picked up by millions of existing dBASE programmers.

Is Nantucket's incremental upgrade strategy working? Most of the Clipper users I meet fall into one of two camps. One camp is using 5.01 for the automatic VMM system alone - ignoring all the additional features of the language I have been discussing. The other group is moving heavily into the new features; pushing the pre-processor to the limits, using 2 MB to 4 MB of VM space for variables, using object-oriented add-ons, abusing code blocks and making full use of the new array features. Admittedly the number in camp two is relatively small, but it is growing in both size and momentum. So the softly, softly approach does seem to be working.

EXE

```
#include "SuperCls.ch"                      // remove this line

Class CargoScreen inherit from saveScreen   // remove the "From"
   var Cargo

// Remove from here

   Message Init( t, r, b, l, data )

end class

// top here

Method init( t, l, b, r, data )

   :Super():init(t, l, b, r)        // Call the superclass init()

// super:init( t, l, b, r)  is a possible future syntax for 5.X

   Self:Cargo := data
return( self )
```

Figure 4 - Single inheritance in Clipper

Chris Sennitt is a Director of ChyDale Software, suppliers of SuperClass and other Clipper add-ons. Contact ChyDale on 0977 683 296.

FREE
3 YEAR WARRANTY

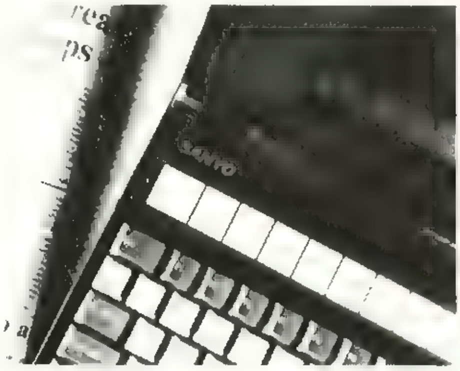
THRIPPLEWOODS

FAMOUS FOR VALUE IN MICRO COMPUTERS

Super SANYO

60 Mb

Notebook17NB6



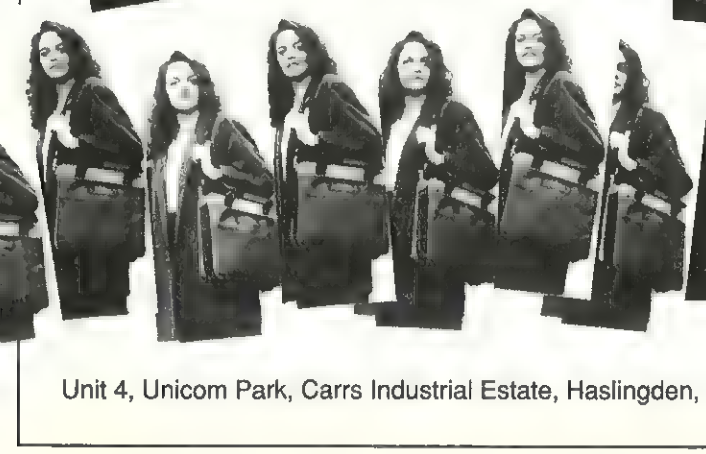
- 286 - 60 MB Hard Drive
- 12.5 MHz Zero wait state
- 1Mb RAM Upgradeable to 5MB

Super Price

£1099 + VAT

- Free PC Tools V.6 (with upgrade to V7.1 for £20)
- Canon BJ10e only £199 + VAT
- *ALSO AVAILABLE* 18NB6 386SX 60 MByte only £1389 + VAT

FREE
CARRYING CASE



TELEPHONE

0706 832005
071 265 4565

CIRCLE NO. 510

Unit 4, Unicom Park, Carrs Industrial Estate, Haslingden, Rossendale, Lancs. BB4 5HU Tel: 0706 832005 Fax: 0706 832100

Configuring programs

Configuration files can make programs more flexible and easier to maintain. Peter Collinson has a neat way of reading them.

There's always a temptation to burn too much knowledge into programs. By this I mean that many programs on UNIX have some information about their environment planted immovably into the binary. For example, programs that do clever things with terminals will need to access the `terminfo` or `termcap` databases; programs that validate users will know about the location of the password file; the wrapper for the C compiler `cc` will know where the binaries of its various passes can found on the system. The list goes on.

It has always been the practice to plant some sensible defaults into the binary and then supply the ability to change them by using program arguments. This is OK for programs that are run seldom, or for the occasion where the change in the defaults is temporary. It's painful to be forced to supply parameters every time a program is called. It's easy to do using aliases in the shell or front-end scripts, but it should be possible to provide better mechanisms.

Using the environment

One way of providing a program with information is to use the environment. This is a set of strings that are passed into processes on the `exec` system call. By convention the strings have the form:

```
keyword=value
```

like

```
SHELL=/bin/sh
```

Users generally set their environment from their shell.

It's easy to access the environment from a program, a standard routine is used:

```
shell = getenv("SHELL");
```

will return the right-hand side string if it exists, otherwise it will return `NULL`.

Setting the environment from programs used to be tricky, but many systems are adopting the `putenv` routine making setting things up somewhat easier. To change the `SHELL` variable, you would say:

```
rv = putenv("SHELL=/bin/csh");
```

The return value is a success/failure indicator.

The new environment variable is passed into any processes that are created by the program that called `putenv`. It's a common mistake to think that the strings are stored somewhere central. They are passed from parent to child. Consequently, the parent can affect the setting of a child's environment but the converse is never true.

However, we do sometimes want to allow a child of a shell to set a variable and a commonly asked question is how this may be done. It needs a slightly cunning trick. The programmer arranges to output a string looking like a shell command:

```
VAR=value; export VAR
```

for Bourne shell and

```
setenv VAR value
```

for `csh`. These strings are printed on standard output. Let's assume that the program that generates this is called `prog`. We want to execute these setup commands in the *current* shell to load the variable. This makes any new command started from the shell contain `VAR` in its environment.

Both flavours of shell contain a command `eval`. The command executes its arguments in the current shell. For Bourne shell, we can say:

```
$ eval "NEWVAL=newvalue"
```

this executes the command

```
NEWVAL=newvalue
```

in the current shell. Of course, this is more useful when the argument is a shell variable.

For our purposes, we must translate the standard output from `prog` into parameters for `eval`. This is easy, we use the back-quote operator. Putting this all together:

```
$ eval ‘prog’
```

This runs `prog` and picks up its standard output making arguments for `eval`. The `eval` command executes the arguments, which will set the environment in the current shell. We are in business.

Using the environment to provide program configuration information is fine. I am happy for it to contain things that are used regularly, like my search path or the `termcap` information supporting my editor and other visual programs. As I write, my current environment is 1471 bytes and contains 29 variables. Every new process that I start has to build a data structure containing the characters and establish a vector of pointers.

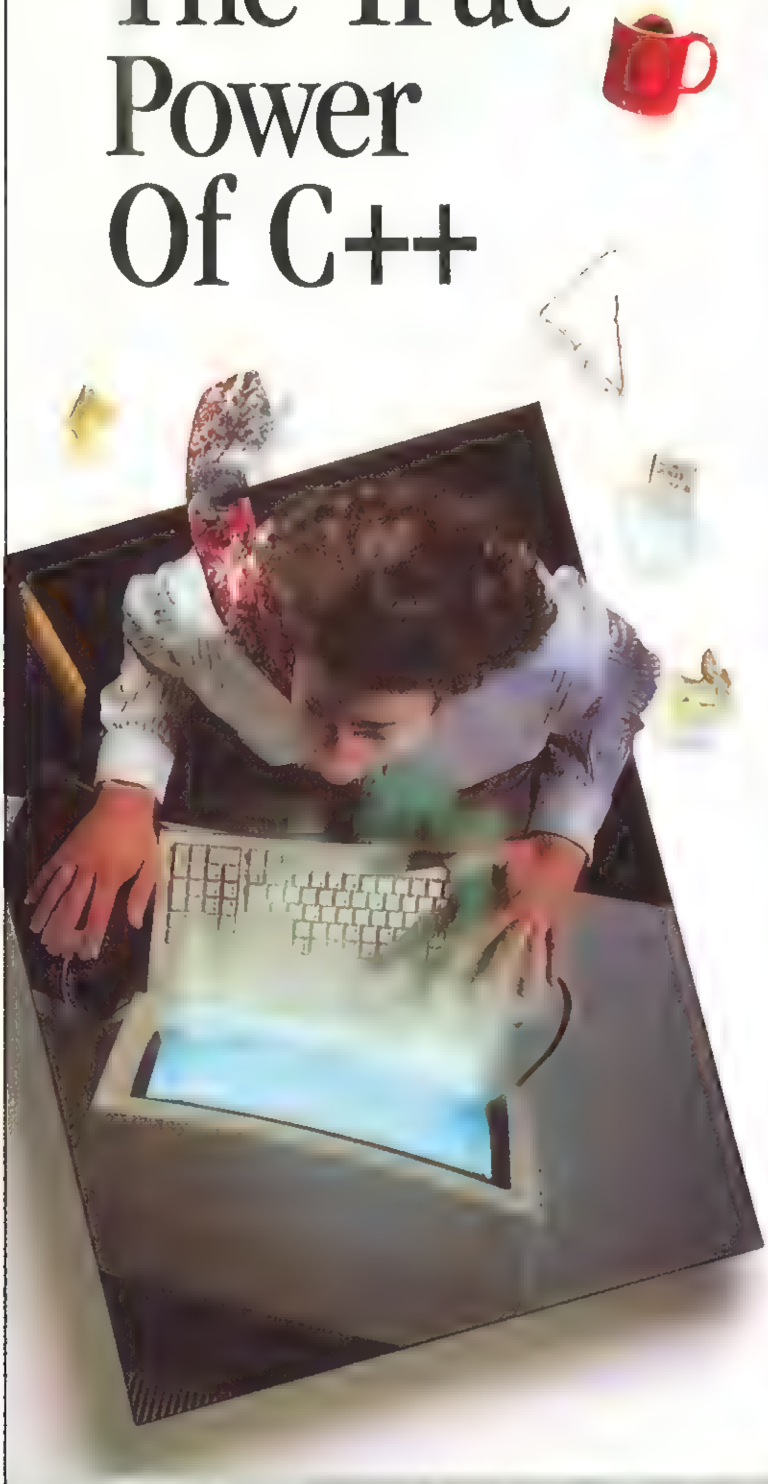
Being of the old school, I resent the CPU wastage. My ears are deaf to the arguments that say CPUs are faster and we have cycles to burn. Too many people say this, and the cycles are being nibbled in all directions. I want to minimise my environment. If your program isn't essential to my work, I don't want its variables cluttering up my world. They take up precious name space and slow me down.

Lists

The data that you want to get into the application may be unsuitable for loading as either arguments or environment variables. In this situation, the approach usually used on UNIX is to read data from a configuration file. The file will most often contain text and is maintained using the normal set of UNIX tools, created with an editor, searched with `grep`, sorted with `sort` and so on.

The simplest type of configuration information is a list of some kind. For instance, you want to send mail to a number of different

Unleash The True Power Of C++



It Takes More Than The Language.

Unleashing the full potential of C + + takes powerful object-oriented tools and a rich library of classes. Using only a C + + compiler and traditional tools yields traditional results – slipped schedules and maintenance nightmares. Only Objectworks\C + +, Release 2.4, provides the comprehensive development environment that allows you to produce extraordinary results.

Speed Your Development and Maintenance.

Objectworks\C + + provides dynamic, graphical browsers to illuminate the tangle of class relationships, object interactions, and program structure, as no static or textual information possibly can. The unique integration of the C + + source level debugger and interactive browsers ensure that essential information is instantly at your finger tips. And, a broad range of cross references are as close as a point and click. All of these tools aid in the understanding of code, increase reuse, decrease confusion, promote consistency and, ultimately, speed your development and maintenance.

An Open Environment.

Based in the latest AT&T C + + language System, Release 2.1, Objectworks\C + + provides support for SunView and X Windows on Sun platforms. Its open environment also allows you to use your favourite C preprocessor, C compiler, linker, profiler, or source code control system. The sophisticated class libraries included in Objectkit\C + + exploit the natural synergy between the development environment and the extensive reusable class libraries. Objectworks\C + + also allows programmers to use existing makefiles without any modification, saving time and money.

Team Programming.

Objectworks\C + + is the only C + + development environment that allows engineers to work cooperatively, loading and browsing one another's code, without interrupting each other. Programmers simply access an information file containing a description of the code they need, and then they query this description.

A I International are uniquely qualified to help you exploit the benefits of object-oriented technology and C + + programming applicable to your development efforts. If you'd like our technical management brief "Exploiting the Full Benefits of C + + Objects", or are ready to unleash the true power of C + +, call us today at 0442 876448

Without Objectworks, you're without objects.

AII

A I INTERNATIONAL LIMITED

The Chapel, Park View House, 1 Park View Road, Berkhamsted, Herts, HP4 3EY
Telephone: 0442 876448 Fax: 0442 877997

CIRCLE NO. 500

We'll teach you to love Jeremy Beadle.

A PC that appears to have a virus. A hard disk that claims to have lost all its files. A backup that won't restore. A corrupted file that locks every user out of the LAN. A salesman who leaves the company, taking a copy of all your customer lists and personnel database with him.

If you keep confidential or business-critical data on PCs, any of these disasters could happen to you.

If you don't protect your company's crucial information, it's all too easy to lose it.

"Data Protection and Security for Personal Computers" is a unique guide for managers and senior support staff. It tells you everything you need to know about how to safeguard the information held on all your company's PCs, including:

- How to avoid viruses.
- How to ensure that backups are taken regularly.
- Preventing unauthorised use of company PCs.
- Buying insurance against PC data loss.
- How to detect and prevent theft of corporate data.
- Windows 3.0 security risks.

"Data Protection and Security for Personal Computers" is written by Robert Schifreen, one of the country's most experienced PC security experts. It costs £145, and there's a 14-day money-back guarantee. Return the coupon below or telephone us now, and we'll send you a free brochure and a detailed summary.

Meanwhile, the next time your hard disk pretends to be empty, or you can't find your backup disks, it'll just be someone having a practical joke.

If you're lucky.

TTK
TECHNICAL PUBLICATIONS

I'm definitely not game for a laugh, especially where my company's data is concerned. Please send me:

□ Full information on *"Data Protection and Security for Personal Computers"*.

□ A copy of *"Data Protection and Security for Personal Computers"* at £145.
Please quote our order number: ____________________

□ Details of your consultancy services on: protection of existing PCs; disaster recovery planning service; detecting hackers; buying and installing secure PCs; implementing a security policy.

Name: ____________________ **Job Title:** ____________________

Company: ____________________ **Address:** ____________________

____

Postcode: ____________________ **Telephone:** ____________________

.EXE 2/92

Please return to TTK Technical Publications Ltd, 10 Barley Mow Passage, London W4 4PH
Telephone: 081 995 9845. Facsimile: 081 994 1533.

CIRCLE NO. 501

people when your program has completed. Conventionally, a list like this is entered with one entry per text line. The program will open the file and input the data a line at a time for storage in a structure, or directly executing a mail command.

The code should worry about ignoring white space at the start of the line and also trailing white space. Some editors will add extraneous spaces at the end of lines, and the person doing the editing will not be aware that it is there. People expect to be able to insert dead space so that their control files look nicer.

Also, it's a good idea to provide a comment mechanism in the data file. I generally use hash ('#') at the start of the line to mean that all the text up to and including the next newline should be ignored. When you are writing the routines to process this type of data file, you will often read data a line at time using `fgets()`. As a result, comments are free - you just look at the first character and loop to get the next line. Incidentally, you can easily ignore blank lines by using the same mechanism.

The ability to have empty lines and comments in a control file adds a lot to its maintainability. Comments will allow the user to describe the values in the file. Also, they can contain version information making the file manageable by a source code control system like SCCS or RCS. A valuable feature is the ability to insert a '#' at the start of line to temporarily remove values without losing the data. Comments are just a good thing.

Simple configuration files

It's often the case that configuration information is relational. Rather than just having a simple list, we will want to have a number of keywords, each taking some value. It's possible that we want to associate more than one value with each keyword.

For example, there is a file used for storing credit in the UKnet accounting suite that I was involved with recently. If a site incurs a bill of less than £10, then an invoice of that sum is sent and the resulting credit stored. Next quarter, the credit is applied to the next bill.

The file that drives this contains lines of data, with one line for each creditor site. The line holds the name of the site, the date that the last charge was made and the amount of money that the site is owed. I use colons to separate the different fields in the data file, but any separator will do.

The credit file is generated automatically as a side effect of running the bill generation program, but it is still text. It can be examined using an editor. It can be updated easily too. If the site is invoiced outside the normal system, a human may want to come along and edit the file to reflect this.

It may seem that reading and processing one of these files is difficult. It isn't. Let's assume that we are writing an interface to the mail system. It's nice to have a private file of aliases so the user can say:

```
mail peter
```

and have `peter` expanded to the real address of the person. We will have a configuration file containing several lines like:

```
peter:pc@hillside.co.uk
```

Each entry in the alias file contains two colon separated fields. The first field is the name that you want to find and the second is the string that is returned for that key.

When implementing this, the initial thought is to use `fgets` to read each line and then examine the first part of the line with `strcmp` to find a matching key. This approach works but is slow, involving enormous amounts of copying. You can use a finite state machine to good effect here. The routine is to be found in Figure 1.

The idea is to loop while reading characters. Different tests will be done on each character that is read depending on the value of the variable `state`. We know where we are in the file, and what we expect to see next. We use our knowledge

```
/* Using a finite state machine */
/* to parse the alias file      */

#include <stdio.h>

char *
readalias(FILE *fi, char *key)
{
  enum {
    NAME, ADDR, SKIPEOL
  } state;

  static char line[BUFSIZ];
  char *pt;
  char c;

  state = NAME;
  pt = line;
  while ((c = getc(fi)) != EOF) {

    switch (state) {

    case NAME:
      if (c == '#')
        state = SKIPEOL;
      else
      if (c == ':') {
        *pt = '\0';
        if (strcmp(line, key) == 0) {
          pt = line;
          state = ADDR;
        }
        else
          state = SKIPEOL;
      }
      else
        *pt++ = c;
      break;

    case ADDR:
      /* success ? */
      if (c == '\n') {
        *pt = '\0';
        return line;
      }
      else
        *pt++ = c;
      break;

    case SKIPEOL:
      if (c == '\n') {
        pt = line;
        state = NAME;
      }
      break;
    }
  }
  return NULL;
}
```

Figure 1 - Code for finite state machine

of the syntax of the file to minimise the processing that is done.

The first state, NAME, is used when we are processing the left-hand side of each line. This is the initial state too. The idea here is to store the incoming characters until we find the colon that marks the end of the first field. When we get a colon, we can compare the data we have captured with the key. If the key matches, then we will need to store the right-hand side because that is the data that we need. We enter the ADDR state to do this. If the key does not match, then we can ignore the remainder of the line entering SKIPEOL state. If we find a hash character, then the line is a comment and should also be ignored.

The second state, ADDR, is only entered when we have matched the left-hand side. We store the characters that we need until a newline is found. Then we exit bearing a pointer to the line.

The final state, SKIPEOL simply discards the character until a newline is found, when it resets the state machine back to the beginning.

The code is actually very rudimentary and if you are planning to use it in anger, you need to worry about syntactically incorrect input. Getting a newline when in the NAME state is currently bad news. The state machine is possibly a little simple and it might be a good idea to deal with leading and trailing white space. You tend to do this by adding more states.

You can also improve on the raw use of strcmp. When dealing with strings I nearly always compare initial characters before executing the costly string comparison. It depends on the data involved, but it can save an enormous amount of processing.

However, apart from this, if you look again at the code you should see that it is very fast. The code path for processing each character is minimal, perhaps three or four tests and a store. Remember also that getc is an inline macro and most of the time is simply picking up a character from a buffer.

Sometimes finite state machines like this can get too complicated, such that it is difficult to see what is happening. I like them for simple applications like this one. They mean that the code works quickly and it is easy to see what will happen in any particular circumstance.

Configuring program suites

When you build a number of programs that work together it's often the case that they all need access to the same files containing various types of information. Rather than burning all the file names into each binary you might think about controlling the programs by the use of a single configuration file.

There are many benefits. All the programs only need to know about the location of a single file and derive all other information from that. It makes it easy to establish parallel environments for testing, and means that you only edit one file if you move the location of a file referenced by the suite.

The configuration file also becomes a handy place to store values that might otherwise be constants in the programs. If the working life of a suite is five or so years, then it's often the case that constants will change. For example, the 'constant' VAT rate recently changed from an integral 15% to a floating point 17.5%.

I generally create configuration files using the familiar

```
keyword: value
```

syntax, and will allow hash comments and blank lines. Different programs in the suite will need different information from the configuration file. The idea is that each program will search for a keyword, or a number of keywords, and ignore anything that isn't relevant to it. The value can be a number or a string - it's just some characters that the program must read and interpret.

The value is deemed to end when the newline is reached. The next line will contain another keyword and value pair. This is often restrictive since long character lines are a pain to maintain. Most of the databases that I construct allow you to continue the data on the next line:

```
# long line
Long: This is a
 long
 long
 line
```

Continuation is signalled by starting the line with white space - either a space or tab character. The reading routine will convert any newline characters into spaces and will consume any white space at the start of continuation lines. As a result the program will see the long line as:

```
Long: This is a long long line
```

There is one further trick that is worth describing. Often a program will deal similarly with several different objects. If your program deals with different types of fruit, then you might have a line saying:

```
Fruit: apple orange
```

Each of these entities might have a characteristic that the program needs to analyse. If we wanted to talk about the colour of the fruit, I would add lines like:

```
apple-colour: red
orange-colour: orange
```

The program dealing with this doesn't know about `apple` or `orange`. It only knows that it is making an internal structure and should read a list of possible fruits from the `Fruit` line. Each of the space separated entities will add a new category of fruit to be dealt with. Additional information about that category can be supplied by looking for keywords starting with the category name. We can generate a new keyword using a `sprintf` format of say:

```
"%s-colour"
```

Additional lookups are now made to get the values of `apple-colour` and `orange-colour`. This scheme is very extensible. Adding a new category, say `lemon`, will not involve any program recoding. You simply change the `Fruit` line and add new keywords:

```
Fruit: apple orange lemon

apple-colour: red
orange-colour: orange
lemon-colour: yellow
```

If you create a new program that needs to know something different about the fruit, then it's easy to add a range of keywords derived from the `Fruit` line.

Of course, in real life, programs rarely deal with fruit, but I hope that you get the idea.

EXE

Peter Collinson is a freelance consultant specialising in UNIX. He can be reached electronically as `pc@hillside.co.uk` *(although your mailer might be happier to put the address the other way round) or by phone on 0227 761824.*

BITON
THE
CLIPPER ORACLE LIBRARY

New for April '92 the FoxPro 2.0 Oracle Connections

CLIPPER - BITON - ORACLE GIVES ACCESS FROM PC TO ORACLE ON - MAINFRAME - MINI - OS/2 OR STANDALONE PC.

INTERESTED IN BITON?

RING: UK (0727) 50658   FAX: UK (O727) 830111

CLIPPER IS A TRADEMARK OF THE NANTUCKET CORPORATION. ORACLE IS A TRADEMARK OF THE ORACLE CORPORATION.

CIRCLE NO. 506

Books

Just to show we're not completely xenophobic C freaks - here's a PROLOG book.

Well, I do declare!

PROLOG - From Formal Specification to Prototype. After reading the title, you might be forgiven for expecting this to be an advanced PROLOG text, but it isn't. As the blurb on the back cover explains, the purpose of the book is to introduce the procedural programmer to the ideas of declarative programming. That's rather a mouthful, but it means that if you know a language such as C or Pascal, and you want to learn more about PROLOG, then this book should be for you.

As a book for PROLOG beginners, there are some rather useful features included in the text. There is a liberal sprinkling of exercises, along with solutions in an appendix. Throughout there are worked examples and 'Practical Activities'. These are mostly practical PROLOG sessions of the 'type this' variety, along with a commentary to explain the results. What if you don't have a PROLOG system? Well, for a few pounds, the authors will send you a PC disk. On it will be a 'no-frills' PROLOG interpreter for MS-DOS, as well as many examples of PROLOG code from the text. As a bonus, the ubiquitous EMACS editor has been included. Although the interpreter was a little fragile when running on my system, it does give you the opportunity to see the book's examples running, and at very little cost. This forms a valuable adjunct to the book.

The book's introduction to PROLOG is unusual, and perhaps startling. All programmers should be familiar with formal specification of an operation. Such a specification will usually include a list of parameters, a method, and a description of pre-conditions and post-conditions. Fewer programmers will be familiar with the notion of specifying the complete operation by these conditions alone. In the book, we learn that if these conditions are specified using PROLOG, then you don't need to worry about the functional details of the operation; PROLOG will do the rest. In other words, if you exactly specify the pre- and post-conditions of an operation, PROLOG will find values to satisfy those conditions. There is no need for the programmer to tell PROLOG *how* to find the values. As the title of the book implies, PROLOG provides the missing link between the formal specification of an operation and a working prototype.

With this blockbuster out of the way, much of the remainder of the book provides what you would expect from any PROLOG text. The bonus is that there is a strong 'hands-on' emphasis. Many (but not all) of PROLOG's primitives are introduced and explained as more complex examples are worked through. However, we are saved from the usual descriptions of expert systems and inference engines. Similarly, there are no travel planners, or draughts playing strategies, although there is a hint of natural language parsing. Sadly, just when you thought it was safe, the very last chapter is devoted to the 'Towers of Hanoi' problem. Yes, I know it is an excellent example of a problem with a succinct recursive solution, but if it is the *only* example, then recursion can't be all that useful, can it...?

How does the book deal with some of the more difficult aspects of PROLOG - backtracking, the cut, and procedural versus declarative meaning, for example? An explanation of backtracking inevitably means resorting to a diagrammatic representation of the CALL-FAIL-REDO-EXIT cycle, and the search tree. A few years ago, the Open University developed an excellent notation for both called the AORTA diagram. All three authors work for the OU, and yet they have not used AORTA diagrams, but less inspired 'engineering' boxes of their own. These work for the examples they have chosen, but for more deeply nested problems, I think there would be trouble. The cut is also explained using diagrams of their own design. Although the basic message 'green cut good, red cut bad' does come across, I would have liked more examples. Procedural versus declarative programming is dealt with by comparing Pascal and PROLOG implementations of the same problems. This links nicely with the notion of using PROLOG for formal specifications described earlier.

It is almost inevitable these days for such a book to include a chapter entitled 'Object Oriented PROLOG'. Accompanying this on the disk is OBLOG. What a tantalising but frustrating glimpse this is. I was certainly left wanting more.

In summary, if you want to learn PROLOG, this isn't a bad place to start. You'll frequently be left wanting to go further, and to do that, you'll need one of the reference texts. If you're an experienced PROLOG programmer, there's little here of interest.

Review by Dr Gareth Blower

Title: PROLOG - From Formal Specification to Prototype
Authors: Peter Leadbetter, Peter Thomas & Ray Weedon
Publisher: Blackwell Scientific Publications
Price: £17.50
Pages: 384
ISBN: 0-632-03161-1

Books Received This Month

Designing Your System with SmartWare II by M Gandoff & M Hicks	Butterworth	£19.95	ISBN: 0-750-6-0425-5	pp330
The Design of OS/2 by HM Deitel & MS Kogan	Addison Wesley	£32.25	ISBN: 0-201-54889-5	pp389
The Standard C Library by PJ Plauger	Prentice Hall	£29.95	ISBN: 0-13-131509-9	pp498
DECnet Phase V - An OSI Implementation by J Martin & J Leben	Prentice Hall	£30.95	ISBN: 0-13-203118-3	pp572
Programs and Data Structures in C (2nd Ed) by L Ammeraal	Wiley	£14.95	ISBN: 0-471-93123-3	pp272

Let Me Explain

In the first of a new regular slot, Francis Glassborow tells us all about the C Users Group.

Let me start by introducing myself. I am the Editor of CVu which is the journal of the C Users Group (UK) as well as being the Chairperson of that organisation. Unlike many editors of Special Interest Group publications I receive more material than the group can afford to publish.

I do not intend to fill the space that the Editor has provided with never-ending reasons why you, the reader, should join CUG(UK). Nor do I intend to waste your time and my space by listing events from our calendar unless they are of more general interest. This time I will tell you a little about us and our aims. In future I will tackle items that should interest programmers using C and its successors. I hope that some of what I write will be of a more general interest.

Let me tackle our title. The 'C' includes C++, Objective C, Concurrent C and other current and future developments in this area of programming. We do not believe that C is in some way special, just that it is an excellent language for many purposes.

The 'Users' includes the entire scale from enthusiastic amateur just starting to the long-term experienced professional programmer. It also includes those that provide the tools that we use, those that offer training and those that develop and monitor standards.'Group' is one of those all embracing terms that avoids some of the overtones associated with 'Club', 'Association' etc. Some aspects of the group do resemble those of a club. Members want to share their interest and insights, they enjoy the sense of belonging and indulge themselves with disagreements. On the other hand members are supportive of one another, they are willing to share experiences and insights and to provide help and advice when they can.

All of us need support occasionally. It is rarely as severe as the young programmer who rang me last year in despair because he could not see how to meet his employer's demands within the time-scale set. A quick hunt round and I was able to point him towards a UNIX tool that achieved the desired result in half the time allowed. The sense of someone else caring and supporting him in his moment of need was much more valuable.

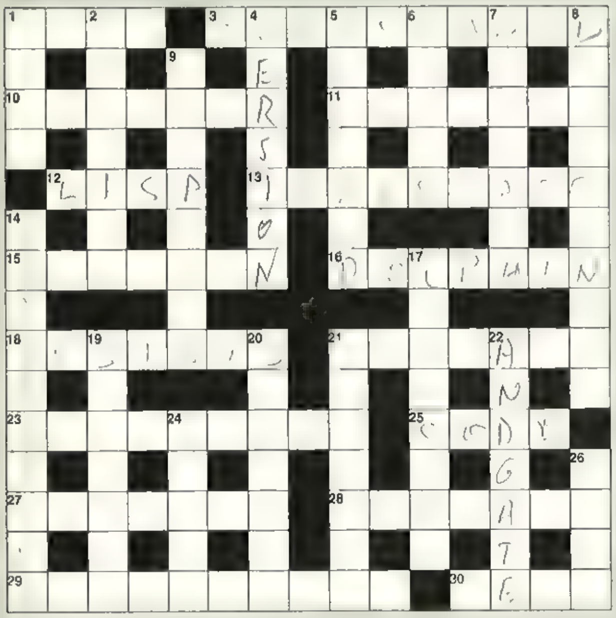
The '(UK)' is simply to distinguish us from an American organisation that uses the same initials.

CUG(UK) has existed in its present guise since 1987. Its primary aim is to support members in developing their programming skills along sound but not dogmatic lines.

EXE

For information about CUG(UK) write to 64 Southfield Road, Oxford, OX4 1PA or ring 0865 246490. Individual membership is £12 and £50 for company membership.

FEBRUARY .EXEWORD



ACROSS

1 Nearly all applied solid-state physics (4)

3 Suffering at work from too large a current (10)

10 Container in backward phone measures may digitise (7)

11 Source of electrons in gun (7)

12 Thpeak thtrangely in some language (4)

13 Erase an eprom (9)

15 Crappy age where is many a game (7)

16 Aquatic mammal worker may block the leak (7)

18 How the conductor opposes the current (7)

21 He charged French dish (7)

23 In banking or police computer maybe a line of ...(9)

25 ... cryptic program? (4)

27 Opening character in one confused ending (7)

28 What n and s do (7)

29 Milliards of Tebbit's vehicles? (10)

30 International at the end of a coding exercise (4)

DOWN

1 Hide behind a set of bits (4)

2 Where to work on a network (7)

4 Type of system we need the latest of (7)

5 Writes the parts of a file (7)

6 Chosen from a drop tediously poured (5)

7 Jovian moon in a play in a 3D set (7)

8 Insulator with such a constant (10)

9 Voracious creatures in popular game (8)

14 Absolute or indexed, say (10)

17 Assign greetings to Catherine, I hear (8)

19 Format where satellites go? (7)

20 Varnish part of OS with alternating current (7)

21 Become delayed and let free in logic unit (3,4)

22 Needs two positive inputs for success (3,4)

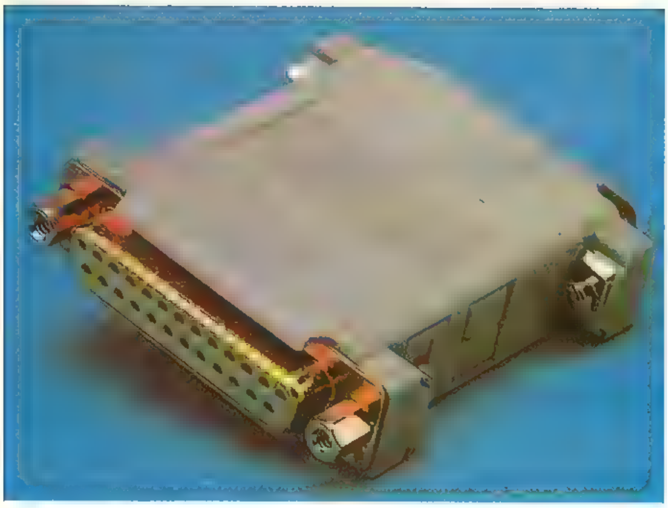
24 The first modern computer (1942) (5)

26 Make no change about 30 (4)

'.EXEWORD' compiled by Eric Deeson

F	E	S	T	I	V	E	■	S	E	A	S	O	N	S
E	■	L	■	D	■	L	■	P	■	S	■	P	■	E
R	E	A	D	Y	■	A	C	O	U	S	T	I	C	S
R	■	V	■	L	■	P	■	U	■	E	■	U	■	S
I	D	E	A	L	I	S	T	S	■	T	E	M	P	I
T	■	■	■	I	■	E	■	E	■	■	■	I	■	O
E	N	L	I	S	T	S	■	S	T	E	P	S	O	N
■	■	I	■	T	■	■	■	■	■	X	■	M	■	■
D	E	B	A	S	E	D	■	F	O	C	U	S	E	D
Y	■	R	■	■	■	E	■	O	■	H	■	■	■	E
N	E	A	T	S	■	N	O	R	M	A	L	I	S	E
A	■	R	■	E	■	T	■	E	■	N	■	A	■	J
M	A	I	N	T	A	I	N	S	■	G	A	M	M	A
I	■	A	■	A	■	S	■	T	■	E	■	B	■	Y
C	O	N	N	E	C	T	■	S	A	D	I	S	T	S

DECEMBER .EXEWORD



How many users of your software paid for it???

SOFTLoK International Limited was established in 1987 with the introduction of our SOFTLoK and SOFTLoK PLUS devices to combat the ever increasing problem of software piracy. Our SOFTLoK range of software protection devices are used by hundreds of software developers from small consultancies to large multinationals. To cope with the ever changing needs of our customers we have developed SOFTLoK II which combines the programmable features of SOFTLoK PLUS with a low unit cost similar to our original SOFTLoK product.

SOFTLoK II™

The Next Generation

SOFTLoK II units are programmable devices containing read/write memory protected by a password. Both the memory and the password can be changed at any time using our routines in your application software. Easy to use menu-driven software is provided to allow small or large batches of SOFTLoK II units to be programmed with their initial data & passwords ready to be sent out with the protected software product.

Price: 1-19 £16.90, 20-49 £15.40, 50+ £13.70

Evaluation kit £20 (SOFTLoK II, manual & software)
All prices excl. VAT and delivery

As SOFTLoK II units plug into the parallel printer port they can be installed or removed in seconds.

- ❑ For IBM PC, PS/2 and compatibles
- ❑ Uses parallel printer port
- ❑ Totally transparent to printer
- ❑ Secure data & password can be changed from your application software
- ❑ Cascadeable
- ❑ 240 bytes of secure read/write memory
- ❑ 8 byte (64 bit) password
- ❑ No programming adaptors required
- ❑ Easy to use SOFTLoK II setup software
- ❑ Routines ready to link with various compilers
- ❑ Easy to follow manual

I-MEX House, 40 Princess Street,
Manchester, M1 6DE, England,
Tel: 061 228 7379 **Fax:** 061 236 6890

CIRCLE NO. 508

Give *Real-Time* Projects a Head Start with the AMX™ low-cost high performance Multitasking Kernel

Features

Fast, reliable operation
Compact and ROMable
PC Peripheral support
DOS file access
C and Assembler supported
Preemptive task scheduler
Time slicing available
Configuration Builder
Intertask messages
Message exchanges
Dynamic operations:
- create/delete tasks, timers
- adjust task priorities
- memory allocation

Buffer Manager	Event Manager
Semaphore Manager	List Manager
Breakpoint Manager	

Sample and Demo Programs
InSight™ Debug Tool for use with most popular debuggers

Targets

AMX 86 for real mode 80x86

AMX 386 for protected 80386

AMX 68000 for 680x0

AMX 80 for Z80, 64180 and 8085

No royalties

Source code included

For more information and a *free* demo disk contact:

ScotWare Ltd
45 Frederick Street
EDINBURGH
EH2 1EP

Tel: 031-225 4858
Fax: 031-220 3550

Real-time and database software specialists

brainy
SYSTEMS (UK) LTD

Motherboards

386SX25 £130 w/cache	£150
386DX33 128K Cache	£265
486DX33 256K Cache	£495

.EXE CLASSIFIED LAUNCH OFFER

80486-33 256KB Cache Mini Twr
4MB RAM - 3.5" + 5.25" Floppy
120 MB Hard Disk
2 Serial 1 Parallel 1 Game Ports
Trident 512 KB VGA Card (1MB option)
UK Keyboard (102 Key)
£1299 exc VAT and Delivery

Call Keith Hickson
Tel: 071 734 5783
Fax: 071 734 7256

Object Oriented Real Time Kernel for TURBO PASCAL

OORTK is an object library for versions 5.5 and 6.0 of Turbo Pascal allowing the application programmer to develop real time and commercial systems using a multitasking approach on PC and PS/2 compatible computers.

OORTK is the first product of its kind for Turbo Pascal and is in use in several countries throughout Europe. Applications produced using OORTK may be distributed royalty free in .EXE form.

OORTK provides multi-threading, multiple prioritized tasks, semaphores, mailboxes, queues, an ADA style rendezvous and many other facilities allowing development of a wide range of applications. The kernel occupies very little memory and the application programmer has total control over the use of system resources.

OORTK costs £97.50 plus £5 delivery (UK) plus VAT. Special terms and prices are available for Educational users.

Dealer enquiries welcome!

For more information write to:
JPS Graphics, 3 Glynne Street,
Queensferry, DEESIDE, CH5 1TA
Telephone or FAX on 0244 821009

All trade marks acknowledged

JPS - Graphics & Microprocessor Systems Design

Offset-Litho PRINTING

Design, Editing, Graphics, Printing & Binding facilities. Fully computerised, we complement your skills and can use your ASCII or DTP files to keep costs low.

- **Manuals**
- **Leaflets**
- **Labels**
- **Stationery**

Fast and Effective Service

Please note our number as this advert does not appear every month

0457 871022

Domino Press
Lawton Square, Delph, Saddleworth OL3 5DT

Version Control

PVCS*	£345
MKS RCS	£155
SMS	£385
Sourcerers App	£169/399
TLib	£95

Editors

PVCS Editor (SPE)*	£175
QEdit	£45
Brief	£189
Epsilon	£169
PVCS Make*	£149
MKS Make	£95
Opus Make	£89

Others

SourceDoc*	£199
PolyLibrarian	£175
Demo II	£175
386/BlueMAX*	£69/£85
SegMentor*	£1695

Readmar Systems
LIMITED

Tel 071 625 5255

*Distributor ACCESS/VISA

KIBWORTH COMPUTER TRAINING

Save time by learning

C or **C++**

with customised tuition in an ideal environment.

Phone or write for details:

68 Springfield Crescent
Kibworth Beauchamp,
Leicester LE8 0LH

Tel: 0533-792653

DCL COMPUTERS
COMPUTER PARTS

MOTHERBOARDS

286 AT 20MHz	£115.00
386 SX 20MHz	£250.00
386 AT 25MHz	£453.00
486 AT 25MHz	£1225.00

Inc. IDE, 2H, 2F, 2 serial parallel interface.

HARD DISKS

40Mb HD	£149.00
100Mb HD	£339.00

Other drives available.

For Price List & Details

Tel: 0902-335632
53 Wells Road, Penn,
Wolverhampton WV4 4BQ

JUST Computer BOOKS

- **Struggling with Sage?**
- **Dicing with DOS 5.0?**
- **Worried by WordPerfect?**

From beginner to expert
We have over 3000 titles to choose from.

Please phone or write for your FREE catalogue

11 Court Rd, Malvern
Worcs. WR14 3EF
☎ 0684-568095

PostScript Library for C
Scripton

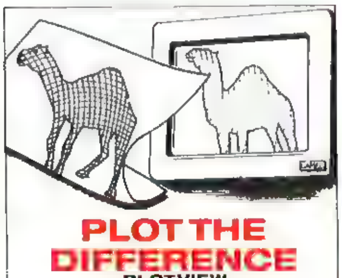
The successful Clipper library is now available for C programmers. There are over 50 functions in Scripton, allowing high definition text and graphics to be produced by PostScript printers. Embed letter-heads, logos and graphics in your Clipper systems.

- Improve report presentation
- Eliminate pre-printed notepaper

Available from
QBS Software
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842

Price: £155.00 + VAT
(including source)

Fax: 081 994 3441



PLOT THE DIFFERENCE

PLOTVIEW

is the simple low cost alternative to paper plots. Able to preview plot images on a graphics screen within seconds.

PLOTVIEW allows paperless plotting and replotting, avoiding delays, improving efficiency.

PLOTVIEW

Has up to 25 separate plot files previewable simultaneously on screen and a zoom function which allows any part of the plot to be inspected in detail.

Easy to integrate PLOTVIEW is ideal for system builders and can be customised to suit their individual needs.

Phone now for more details about PLOTVIEW the versatile tool that can really make a difference to your productivity.

CAMEL SERVICES LTD.
Telephone OXFORD (0865) 512678

BUGS WANTED!

A new BASIC User Group is looking for volunteer regional co-ordinators to help arrange regular meetings around the country.

If you have an interest or any experience with BASIC programming, please write to:

BUG
30 Osprey Close,
West Drayton, Middx UB7 7JE

The Secret of Successful Advertising is.....

CLASSIFIED

To find out how you can reach 17,000 professional developers for as little as £85, call Marc Warren on 081 994 6477

The Babelfish Paradox Data Driver for Clipper Summer '87 and 5.0x versions.

Call Now for a free Demo Disk.

QBS Software Ltd
10 Barley Mow Passage, London W4 4PH
Tel: 081 994 4842
Fax: 081 994 3441

Type & PostScript

PostScript Interpreters: from £4 to £4,000
Fountain, GoScript, Freedom of Press, Ultrascript, PS Tutor, LincPage, NTG2000, Walland Orb

PostScript LaserJet Upgrades from £239. Adobe, HP, Pacific Page, UDP TurboScript

PostScript Utilities: ATM, PSFx, PSPlot, Books

Type 1 Fonts from £3 to £3,000: Public Domain, Adobe, ZSoft, Digi-Duit!, Walland Orb

HP Bitmap Fonts. ZSoft, Typografica, Digi-Duit!, Atech, DTS, Walland Orb, Tailor Made

Laser Printer Memory Upgrades. HP, Canon, IBM

Custom Fonts & Logos. HP Bitmap, Type 1, EPS

Custom HP Font Cartridges: from £90 (10 off)

If you need to print Cyrillic, Turkish, or Classical Greek, or need design consultancy, then call or fax. We have good prices on a sensible range of print and graphics products: we also put together very fast, reasonably priced PCs.

David Pollard Associates

Folly Bridge Workshops, Thames Street,
Oxford OX1 1SU 0865 240048 Fax 792277

ORCHID WORKSTATION

Orchid '486/33 Motherboard
Orchid Fahrenheit Video Controller
NEC MultiSynch 4FG Monitor
Big Black Box, Cherry Keyboard
16 MB, Single Floppy: £2469 +vat
Motherboard + Fahrenheit: £1050

HDD, second FDD, tape streamer
SCSI controller etc. at trade cost +7%

The Orchid has a 32 bit direct video data path and the controller is faster than standard VGA. Compared with an ordinary '486, Windows applications run 18 times faster on the Orchid.

David Pollard Associates

Folly Bridge Workshops, Thames Street,
Oxford OX1 1SU 0865 240048 Fax 792277

DO YOU NEED MORE MEMORY?

We have three libraries to give you just that!

XMEM - Virtual memory Manager using base, Expanded, Extended or Disk. Replaces malloc. **£145**

MEMORY TOOLKIT - Library of routines to manage Expanded and Extended memory plus big DOS arrays. **£95**

EMPTY SHELL - C routines for freeing up memory when shelling to a large program. Uses EMS, XMS or disk to swap **£165**

GREAT WESTERN INSTRUMENTS LTD

Unit 1m Farrington Fields
Farrington Gurney
Bristol BS18 5UU
Tel 0761 452116
Fax 0761 453226
VAT Extra Carriage Free

WORKSTATION MEMORY

SUN 4 COMPATABLE MEMORY		
4/20/SCL	4Mb	£200
SPARCSTATION ELC	4Mb	£200 00
	16Mb	£675
4/40 (IPC) 8ONS	1Mb	£44.00
7ONS	1Mb	£48.00
8ONS	4Mb	£165.00
7ONS	4Mb	£184.00
SPARCSTATION IPX	4Mb	£200.00
	16Mb	£675.00
4/60 SPARCSTATION 1	1Mb	£44.00
	4Mb	£165.00
4/65 SPARCSTATION 1+	1Mb	£44.00
	4Mb	£165 00
4/75 SPARCSTATION 2	4Mb	£165.00
4/110,150	1Mb	£44.00
4/260,280	8Mb	£1870.00
	16Mb	£2300.00
	32Mb	£3995.00
4/330	8Mb	£740 00
	16Mb	£1205 00
	32Mb	£2250.00
	64Mb	£4140.00
4/370,390	4Mb	£165.00
4/470,490	32Mb	£POA
	128Mb	£POA
SUN 3 COMPITABLE MEMORY		
3/50, 3/52	4Mb	£475.00
	8Mb	£765.00
	4MB	£305.00
SMM TOOL KIT		£35.00
3/60	1Mb	£44.00
3/80	1Mb	£44.00
3/75,110,140,150,160,180		
	4Mb	£750.00
	8Mb	£975.00
	12Mb	£1300.00
	16Mb	£1620.00
	20Mb	£1950.00
	24Mb	£2275.00
3/260,280,470,480	8Mb	£1870.00
	16Mb	£2300.00
	32Mb	£3885 00
3E120	4Mb	£505.00
	8Mb	£785.00
	12Mb	1075.00
SUN 386 WORKSTATION MEMORY		
386i,150	1Mb	£44.00
386i,150x,250	1Mb	£44.00
APOLLO DN MODELS		
DN2500 8ONS	1Mb	£44.00
7ONS	1Mb	£48.00
DN3000, DN3010	2Mb	£280.00
DN3010A,3040,3500,3550,4000,4500,5500		
	4Mb	£370.00
	8Mb	£630.00
HP 400 SERIES		
HP9000/400,400dl,400t,425t,400s,433s		
	4Mb	£345.00
	8MB	£740.00
	16Mb	£1335.00
	32Mb	£2540.00
HP APOLLO 9000/425E		
	4Mb	£355.00
	8Mb	£725.00
	16Mb	£1200.00
HP APOLLO 9000/700		
	8Mb	£990 00
	16Mb	£1485.00
	32Mb	£3940.00
IBM COMPATIBLE MEMORY RS6000		
RS/320/320H,520,530,550,730,930,950		
	16Mb	£1250.00
	32Mb	£2290.00
	64Mb	£4390.00
APPLE COMPATABLE MEMORY		
Mac II,IIx,IIcx	1Mb	£42.00
SE,SE/30,PLUS,CLASSIC,		
LC,IIci,IIsi	4Mb	£170.00
Mac IIfx	4Mb	£200.00
	16Mb	£825.00
SILICON GRAPHICS COMPATIBLE		
PERSONAL IRIS 4D/20,25		
	1Mb	£44.00
	4Mb	£165.00
PROFESSIONAL IRIS 4D/50,70,80,85		
	1Mb	£44.00
	4Mb	£165.00
POWER SERIES 4D/120,210,220,240,280,		
310,320,340,380	2Mb	£135.00
NEXT COMPATIBLE MEMORY		
NEXTcube,NEXTstation	1Mb	£42.00
	4Mb	£170.00

SABREBANNER
Tel 0532 854521

VISA, Access & MasterCard Taken
Fax 0532 853026

IMPRESS YOUR FRIENDS!
LOSE THAT FLAB!
IMPROVE YOUR SEX APPEAL!

All good ideas - but you won't achieve any of them by

WRITING FOR .EXE

However, unlike jogging, there could be some beer money in it for you. For a copy of our Contributors' Notes, write to:
The Editor, .EXE Magazine,
10 Barley Mow Passage,
Chiswick, London W4 4PH.

TAKE UP WEIGHT-LIFTING!

Computer MUSIC Systems Ltd

5-7 Buck Street,
London NW1 8NJ.
071-482 5224

Voyetra Sound Factory™

Sound Factory™ is a complete audio development kit for DOS or Windows 3, from Voyetra, the leaders in IBM music software. Containing drivers for a wide range of sound cards, digital audio boards & MIDI interfaces, Sound Factory™ gives developers as wide a market as possible. Call or fax us for further details.

We also stock a wide range of PC software and peripherals.

Specialists in Music Software

PC Data Security Guide

Data Protection and Security for Personal Computers is a guide for support staff and managers. Everything you need to know about:

- Keeping viruses out of your company.
- Drafting and implementing a corporate data **security policy.**
- Persuading staff to take regular **backups** that are known to be reliable.
- Detecting and eradicating **software piracy.**

Includes comprehensive product guide and lots of trouble-shooting advice.

The price of £145 includes 90 days' **free telephone support** from the author.

Please call TTK Technical Publications on 081 995 9845 for details and a **free brochure.**

Clipper PostScript Library
Scripton

There are over 50 functions in Scripton, allowing high definition text and graphics to be produced by PostScript printers. Embed letterheads, logos and graphics in your Clipper systems.

- Improve report presentation
- Eliminate pre-printed notepaper
- Put picture and graphics in reports

Available from **Price: £155.00 + VAT**
QBS Software
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842 Fax: 081 994 3441

windows SHAREWARE

ikonworks takes a different approach to the distribution shareware for Windows .

HIGH DENSITY DISKS
By compressing an average of 20 programs onto HD disks we offer extremely good value for money.

BUSINESS SOFTWARE
We supply a wide range of applications and utilities which are designed to meet the needs of people who use Windows in business.

FREE UPGRADES
Most of our disks can be returned with your next order for a free upgrade.

CUSTOM DISKS
We can build up disks to order from your choice of over 650 applications, utilities and fonts.

TELEPHONE OR WRITE FOR A
FREE CATALOGUE

ikonworks

22 Hope Street
Lancaster
LA1 3BQ
0524 39283

R&R Relational Report Writer

Database reports made easy for dBase, Clipper and compatibles. Also for Paradox, 123s and Quattro Pro.

Turn raw data into useful information using Lotus like commands and menus.

Available from
QBS Software Ltd
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842
Fax: 081 994 3441

Call for free demo disk

PostScript Utilities

PostVue £99 + VAT
A visual de-bugger for PostScript code. Watch your code execute, examine stacks, dictionaries and state variables. Set break points, warning levels, delay times.

PostPP £249 + VAT
Un-scrambles PostScript code and lays it out in human readable form

PPS £49.00 + VAT
For all PostScript printers. Save paper with this versatile driver for ascii files. Output in columns, landscape, choose fonts, headers, line numbering etc

Available from
QBS Software
10 Barley Mow Passage
London W4 4PH
Tel: 081 994 4842 Fax: 081 994 3441

RONICOM
RECRUITMENT

5-7 Sedley Place (off Oxford Street), London W1R 1HH
Tel: 071 491 3640 Fax: 071 499 2546

TEAM LEADER

SW London **up to £25k + Bens**

OSI skills - X25 - X400 - a good coding background in C is essential for this quality software house, a good knowledge of Windows 3 would be of benefit.

SOFTWARE ENGINEER

SW London **up to £19k**

Windows SDK, good calibre C, some networking, X25 in an O.S.I. environment, you should have at least two years experience.

SYSTEMS SOFTWARE ENGINEER

Herts **£neg + Bens**

A quality 4GL software house is looking for a UNIX, C, UNIPLEX background in an X-Windows environment, you must have good DOS skills with shell & kernal ability - perhaps even some X400?

CAD EXPERIENCE

Kent **up to £20k**

Geosecma, Landcadd development and tailoring - possibility of European work.

C++ WINDOWS ASSEMBLER

Berks **up to £20k + Car**

Good development skills are required for this innovative Windows software house; young, extremely competent and enthusiastic please.

OS2 - WINDOWS SDK-SQL

W London **up to £27k**

An intriguing development that will create great prospects, the current products are good and the projected products look like they will do very well.

OSF MOTIF - X WINDOWS - C

Cambs **Good + Bens**

Good development engineers are required for this UNIX Windows software house. Some client contact tailoring is expected, so you should not be too shy an individual.

LOW LEVEL BIOS C

Surrey **£18k + Bens**

BIOS MASM TASM Device Drivers TSRs your bag? Any 80x86 driven assembler, then this PC manufacturer will offer you a bright future!

If any of these positions are at all suitable, or even if your skills are not quite what is required, do not hesitate to give Mike Dearing a call on 071 491 3640 or after hours on 081 767 1003

the soft (corporation)

Specialists in relational databases, open systems and structured techniques

ORACLE/SQL/ EMBEDDED SQL

Candidates with skills in SQL and embedded SQL are immediately required by clients involved in applications as varied as financial information systems to pan European manufacturing/ distribution and leasing systems. Strong SQL experience coupled with low-level language interfaces to 3GL's and screen painting methods are important. Training in the latest Oracle tools and methods given where necessary. The new and existing implementations are running under VMS, ULTRIX, UNIX and sunOS with some links to multi-user PC systems. Other database knowledge may also be of interest.

£12,000-35,000+
Ref SC/1/ exe

INGRES

Excellent opportunities exist for any INGRES development staff from Programmers up to Project Manager level. Candidates will be using the most advanced development tools available with companies investing in leading edge technology. INGRES/STAR, GUI's, SQL and OPEN SQL, decision support/application development tools, user interfaces and structured methods (SSADM, YOURDON, JACKSON), C, COBOL, are the priorities Potential to work across varied hardware platforms is also a preference. Opportunities exist in Banks, Building Societies, Software and Systems Houses and Governmental Bodies

£12,000-35,000+
Ref SC/2/.exe

SOFTWARE DEVELOPERS DATABASE DESIGN

SYBASE, INGRES, INFORMIX,ORACLE, UNIFY,CLIPPER, PRO IV, POWERHOUSE, RDB, ALL, SEACHANGE, PROGRESS, SB+, INFORMATION-and interfaces to 3GL's - C, C++, PASCAL and FORTRAN and even COBOL and BASIC! Our client companies require candidates with the following skills: Technical Software Development, Software Engineenng, Pre/Post Sales, Database Design and Systems Consultancy Areas of particular interest include: Distributed Systems, Object Oriented Databases, LANS, WANS, Graphics Image Processing, Windows, Formal Design Methodologies, CASE Tools, Expert Systems, Knowledge Engineering, Office Automation, Compilers, Low Level Design, Data Integration and varied application design. COMPUTER POLYGLOTS need apply!

£12,000-40,000
Ref SC/6/ exe

C/UNIX/MS-DOS

Software Houses and End Users in Manufacturing, Commercial, Scientific and Government application environments alike require excellent C skills. Low-level machine knowledge, operating systems internals, and use of debuggers/compilers are required. Software development experience is the key, and being able to deliver high performance, high quality, well specified software in competitive time scales. Opportunities vary from small software companies involved in expert systems, GUI's, Images Processing, GIS,EIS, Communications, Networking and Object Oriented Databases to large scale communication companies systems. Graduates through to senior software engineers/team leaders are required. UNIX internals and Kernel level knowledge are also very much in demand.

£14,000-35,000
Ref SC/3/.exe

X WINDOWS/ MS-WINDOWS/MOTIF

Graduates (1 year+) to Senior Software Engineers with an interest in advanced development environments need apply for varied positions with companies dedicated to leading edge technology. A mixture of the following skills in a development environment are preferred: GUI's, Document Image Processing, OCR technology, Client/Server applications and WYSIWYG techniques coupled with experience of UNIX, sunOS, MS-DOS, or ULTRIX/VMS and Interfaces to relational (primarily INGRES or SYBASE) and Networking (TCP/IP, NFS, X400) technology. Application areas vary from Retail, distribution to finance and Software Vendors to End User environments

£12,000-35,000 + BENEFITS
Ref SC/5/ exe

SYBASE/UNIFACE

Due to the continued expansion of the market for RDBMS as a multi purpose software development tool and particularly the adaptability of SYBASE, its popularity is growing in manufacturing, scientific, industrial, commercial and financial application areas. Clients throughout the UK and Europe require SYBASE experience in a VMS, UNIX and even PC environment. Again interfaces to 3GLs, Structured Methods and Communications Protocols are also important. Please call to discuss the varied opportunities at all levels, including setting up new development departments.

£12,000-40,000 + BENEFITS
Ref SC/8/ exe

COMMUNICATIONS/ NETWORKING

Developments as varied as Voice-activated relational database recognition systems, Protocol enhancement at transport, session and presentation level and World-wide communications systems utilising LAN's and WAN's across different hardware platforms are currently available. Experience of ETHERNET, TCP/IP, NFS, X25, X400, X500 in a UNIX, VMS, sunOS and also fault tolerant environments are required. Some exposure to structured methods and other leading edge technology would be a bonus, though training will be given. Knowledge of industry standards and committees is also relevant at more senior levels. UNIX Kernel knowledge is at a premium.

£12,000-£35,000
Ref SC/4/.exe

SOFTWARE ENGINEERS

Clients-End-Users, Software Vendors and Software Houses dedicated to strategic implementation of leading edge technology and integration of applications across different hardware and operating systems platforms require candidates to degree level with a scientific/technical development bias and a 1-3 years' experience. There are two main options:-

TECHNICAL DEVELOPMENT: Continued use of UNIX, VMS, MS-DOS, C, Windows, Pascal, C++, Ada, Prolog, OOPS, Networking and Communications with companies offering technology based careers and management responsibility.

COMMERCIAL DEVELOPMENT: Using technical skills already developed, but offering opportunities to apply analysis and design skills rather than remain a 'technical guru'. Please call to discuss your particular career path, growth and potential

£12,000-25,000 + BENEFITS
Ref SC/7/ exe

INFORMIX/CLIPPER

Clients involved in small to medium database projects (10 to 35 users) utilising products like INFORMIX, CLIPPER, DBASE IV and FOXBASE require Programmers Analyst/ Programmers with 4GLs, C, Pascal, Turbo Pascal, Quickbasic etc... or some of these skills for new and existing developments. Potential to get involved in Networking projects and in some cases larger scale projects involving cross-training also exist.

£12,000-22,000
Ref SC/9/.exe

STOP PRESS

Pre/Post Sales UNIX Systems London.
SQL Systems Consultant - Berk/Surrey/London.
UNIX Systems Administrators - All Areas.

CONTRACT AND PERMANENT POSITIONS ARE UK WIDE, SO PLEASE TO DISCUSS YOUR PARTICULAR SKILLS AND REQUIREMENTS.

The Soft Corporation Chancery House 319 City Road
London EC1V 1LJ Tel: 071 609 5501 Fax: 071 700 5787

Job market flat???... ..shouldn't you be in GIS??

G.I.S. is a rapidly growing application area of I.T. Public and Private sector users are investing heavily in G.I.S., many linking the database information from the G.I.S. with their existing database management systems.

If you are a skilled systems and programming professional with experience of using Oracle, Ingres, Sybase or one of the newer OO DBMS toolsets for developing applications software, you should consider moving into G.I.S. Below are THREE current opportunities which illustrate the point.

Database Analyst, £24000, West Herts

If you have at least 2 years experience of developing database applications, ideally in Ingres under UNIX this world leader in G.I.S. will interest you. Your software will enable clients to fully integrate G.I.S. into their corporate M.I.S.

Analyst Programmer, £22000, Maidenhead

Your skills should reflect a detailed understanding of textual and graphical databases probably gained with a CAD system supplier. You should be fluent in C under both DOS and UNIX.

Software Engineer, £19000, Reading

You will join a team developing a new generation of G.I.S. You should be very familiar with SQL and C. Training in the products will take place in Australia.

To apply for one of the above vacancies or to find out how your skills could be transferred to G.I.S. call **ALAN CARNELL** at Concurrent Appointments.

Concurrent Appointments
Software Recruitment

27 FIELD CLOSE
HARPENDEN
HERTS
Tel: 0582 712976
Fax: 0582 764858

QUESTOR
Computer Personnel

PROGRAMMER **N. Herts c. £12k**
Very highly regarded s'ware house with full order book needs an additional applications programmer, to work under Windows3 & PM. Around 1 year's commercial experience of C or a 4GL necessary. *Contact P.H.*

TECHNICAL AUTHOR **N.Cambs to £20k+ bens**
Leading edge software company seek a technical author with several years' experience of writing with a commercial software bias. Exp' of projects from concept to completion, OS/2 - Windows or AS400 applications will be of added interest. *Contact A.H.*

SENIOR FIRMWARE ENGINEER **Bucks circa £22k**
Experience in high speed image processing using CCD technology, a high level language & 8086 & 68000 assemblers, combined with good PC knowledge. *Contact P.H*

SYSTEMS SOFTWARE ENGINEER **Middx to £26k**
To work on microprocessor-controlled systems. Relevant degree, 3-5 years commercial design/development experience on IBM PC & SUN equip' plus scope & logic analysers. Team Leadership useful. *Contact P.H.*

NETWORK & SYSTEMS SUPPORT **E. Herts £15 - 20k + car**
Growing IBM systems house & VAR require an experienced support analyst with sound knowledge of MS-DOS & Xenix in a networked environment. You will provide telephone and on-site support, system & network installations and network administration for clients on site. *Contact T.S.*

ADA - PROGRAMMERS
ANALYSTS and DESIGNERS **Surrey + others Neg**
With over 2yrs' experience in one of the above areas, you will ideally have experience with structured methods, Naval systems, performance critical s'ware or team leading. *Contact M.D.*

CAD - OPERATOR **Wilts £Neg**
With several years Autocad exp' and a mechanical design background this prestigious company requires good candidates to work in a buildings services

Regal House
55 Bancroft
Hitchin
SG5 1LL
Tel : 0462 438373
Fax: 0462 421272

Evenings:

Alan Hewson
0462 459338

QCP

WEST YORKSHIRE

Trainers for accounting and admin software	*£12,000*
Project Installation Manager (UNIX)	*£14-16,000*
Trainers & Programmers for PROGRESS 4GL/MSDOS & UNIX environments	*£12,000*
4GL Analyst Programmers to work in UNIFACE/SYBASE/ULTRIX environment	*to £15,000*
TAS 4GL Support Programmer	*£10,000*
'C' & 8086 Assembler Programmer	*to £15,000*
Clipper Analyst Programmer with 'C' on Novell	*to £15,000*
Programmer/PC Support, Clipper, dBase or Sycero DB	*£11,000*

For your next career move around West Yorkshire telephone Vincent Atherton on Leeds (0532) 504560 or write to:

AIREDALE RECRUITMENT
Realtex Hs., Micklefield Lane,
Rawdon, Leeds, LS19 6AX

AIREDALE RECRUITMENT

PC Software Localisation & Test Specialists

C : DATABASES : SQL : MS-WINDOWS

Superb opportunity to be a founder member of a specialist team created to localise a leading Windows database system for European markets. Conveniently situated near Oxon/Berks border.

PROJECT MANAGER	**to £35,000**
TEST TEAM LEADER	**to £27,000**
SOFTWARE ENGINEER	**to £24,000**

We are seeking candidates who are familiar with the development and usage of translation tools/kits and with the test processes involved in the translation of baseline products into local languages. Technically, you will have considerable experience of PC Databases, MS-Windows and be fluent in C. Exposure to product release mechanisms in a fast moving software publishing company would be a distinct advantage. Above all, candidates should be graduate calibre, results orientated and enjoy working in a technically stimulating team environment.

If you feel you are capable of rising to the challenge of these existing career opportunities please telephone **David Adcock** on **(0865) 742456/7** or send your CV to **Haybrook Appointments, Suite B, Regal Court, 112 London Road, Headington, Oxford, OX3 9AU.**

Haybrook
APPOINTMENTS

PC Recruitment Specialists

UNIX AND WINDOWS

Unix System Support

South London

£20,000 + car + benefits

This highly successful international hardware vendor are seeking a Unix Systems Support person to be involved with leading edge technology. This opportunity is varied and challenging and covers 2nd line client support, pre-sales, demonstrations and evaluations of new products. Your background will include 2 years experience of *Unix* product support or large systems support and technical fluency with *Unix* to *Kernel* level. You are educated to 'A' level standard and have excellent communication and interpersonal skills. *Ref: PCEX15/1*

Ingres/Oracle Analyst Programmers

London/Surrey

to £26,000 + benefits

A prestigious retail outlet, based in Central London, is currently seeking a senior analyst/programmer to be involved with the new development of in-house systems. With extensive *UNIX* and *Oracle* skills the ideal candidate will be experienced in all phases of the project cycle including structured development and analysis. A solid understanding of database techniques and excellent social and interpersonal skills are fundamental to this role. Alternatively, a leading software house, based in Surrey, urgently require *Ingres* analyst/programmers to be responsible for the development of commercial systems. If you have a minimum of 2 years Ingres under *UNIX*, this company can offer you ongoing training and enhancement of your skills in a stimulating and challenging work environment with genuine prospects for career progression. *Ref: PCEX15/2*

Windows and 'C' Developer

The City

to £20,000 + benefits

This young and rapidly growing financial software house is currently seeking a *Windows* applications developer with at least one years' *SDK* experience. You will have at least two years' *Windows 3* experience and solid 'C' skills gained in a commercial environment. Project work will include full involvement from initial specification through to implementation. Enhancing the Windows experience of the current team, you will apply your knowledge to exciting financial developments and gain valuable experience in the process. This is a genuine opportunity to grow with a dynamic and merit based company.

Ref: PCEX15/3

Oracle Analyst Programmer

Central London

£16-20,000

In line with its intention to implement an open systems strategy, utilising *HP9000 Unix* boxes, this prestigious Central London consultancy currently has a requirement for two Oracle analyst programmers. These positions will be of particular interest to candidates whose ultimate career ambitions lie in the area of consultancy as such options are available once the internal systems migration has been completed. As a result, candidates must combine both strong technical skills – a minimum of 18 months *Oracle* development in a *Unix* environment – with excellent interpersonal and presentation skills and a good academic record. Graduates are preferred. *Ref: PCEX15/4*

For further details of these and other permanent vacancies, please contact S. Atkinson, quoting the relevant reference, on 071-734 4010 (office hours) or 071-431 2316 (evenings/weekends). Alternatively, write to McGregor Boyall, Sutherland House, 5/6 Argyll Street, London W1V 1AD, or fax your CV on 071-734 1297.

mcgregor
boyall

IT HUMAN RESOURCING

FOR PERMANENT AND CONTRACT IT RECRUITMENT, TELEPHONE 071-734 4010

ASH ASSOCIATES

We specialise in the Recruitment of Software Design and Support Engineers in the South East for Real Time Applications including GRAPHICS, COMMUNICATIONS, CONTROL / ROBOTICS, SIGNAL/IMAGE PROCESSING & MODELLING.

NEW PRODUCT DEVELOPMENTS

Medical Electronics Surrey £22K

This High Technology company is Now seeking to expand it's Software Team working on New Projects for Image Processing & Scientific Instrumentation Systems. You will have at least 4yrs software experience particularly with C and UNIX. Experience of Windows and Low Level Graphics also useful.

Office Automation Berks £20K

This well known designer of computer systems seeks an experienced Software Engineer to be responsible for developing their New Projects in office communications. 3yrs plus software design is needed of both High and Low Level languages (C, Intel Asm) gained in a PC-comms environment.

Control Systems Surrey £18-24K

Expanding designer of Real Time control systems seeks experienced software design engineers at all levels to Team Leader to work on their New developments. Your 2yrs plus experience of Real Time and Low Level software design (C and Intel/Motorola assemblers) combined with the capability to lead a Team of engineers will be most useful.

Call James Hunt or Ron Cook NOW!
TEL: (0425) 475480 (24hrs) FAX: (0425) 480807

ASH associates
Recruitment Consultants
3 Pipers Ash, Ringwood, Hants, BH24 1UF

Herts **to £25k**

Developing integrated corporate office software for the world-wide market place. You should have at least two of the following: **'C', UNIX, TCP/IP, X-Windows, GUI, E-Mail** or **Graphics.**

Bucks/Berks borders **to £23k**

Revolutionary developers of emulation technology seeking self-motivated software engineers for full project life-cycle development. If you have strong 'C' skills combined with some of: **UNIX, VMS, MacOS** or **MS-DOS**. 1st class employer offering profit share, share options and flexitime.

Yorks **£15k - £20k**

Forefront Spatial Information Developer is seeking innovative software designers with sound problem solving skills. Your skills should include 2 years of **'C'/UNIX** applications and experience of using relational database tools, any **structured methods, graphics** or **Windowing.**

City **to £30k + benefits**

Major financial institution is seeking bright software staff to develop **Real-Time Market Information Systems** covering such applications such as currency options, price modelling, risk analysis and time series databases. You should have solid 'C' skills combined with some of: **UNIX, VAX/VMS, Ethernet, C++, OS/2, Sybase, Windows, Motif** or **4GL's.** Benefits include subsidised mortgage!

MANY OTHER OPPORTUNITIES THROUGHOUT UK AND EUROPE

071 371-8496
ACUMEN SEARCH & SELECTION INTERNATIONAL
EBC House, 1a Ranelagh Gardens,
London SW6 3PA Fax: 071 371-8502
Out of hours telephone: 081 780-0637

ACUMEN

C Programmers

Hertfordshire *£Neg.*

A challenging opportunity to work in the R & D section developing new technologies for this International Group. You will be Degree or M.Sc qualified & have at least 12 months experience of coding in C. A knowledge of MS-Windows is desirable but not essential. This position is ideal for a programming purest who flourishes on coding challenges. You will, in return, receive a good salary, work in a state-of-the-art environment & use the latest software and hardware.

Contact: Paul Innes Ref: PI0601

Design in Colour

Herts *£12-23K*

Leading the development of Unix based computer controlled Workstation/Imaging systems for the Graphics Industry, our client is seeking Software & Software Support Engineers with skills in C, Unix, Graphics, X-Windows or Macintosh. You will play a key role in product development for the commercial market worldwide. If your experience includes any of the above, please call for a confidential discussion.

Contact Teresa Maddern Ref: TM0601

GUI Specialists/Oracle

W. London/Midlands *to £25K*

This expanding, successful, communications specialist have an exciting new MIS project & require GUI specialists preferably with Apple Macintosh or ORACLE experience. There are excellent career opportunities with a comprehensive training scheme & generous benefits package.

Contact Christine Trybus Ref: CT0601

Systems Programmers

Oxfordshire *to £18K*

An opening now exists for an enthusiastic programmer to develop complete Turn Key Solutions of the future. A good degree in Computer Science and some experience in each of the following categories is required:

Languages: C, Pascal, Assembler
Operating Systems: DOS, UNIX, XENIX
Tools: WP, Databases (particularly Paradox), Spreadsheets
Communications: ARCNET, NETBIOS, NOVEL, TCP /IP
Experience of programming, device drivers & H/W interfaces.

Contact Paul Innes Ref: PI0602

C/Unix/XWindows/RDBMS/SQL

London/Cambs *to £30K*

We have urgent requirements for two successful reputable service organisations to provide consultancy & develop financial & commercial applications using C, UNIX & XWINDOWS & INGRES/ORACLE. The successful candidates will be educated to degree level with at least 2 years experience with C & UNIX & either X WINDOWS or RDBMS/SQL.

Contact Christine Trybus Ref: CT0602

For more information call **0442 231691** days or 0442 69740 eves/wkends. Alternatively send your CV to the address below or fax details to 0442 230063

Executive Recruitment Services
Hempstead House, Selden Hill,
Hemel Hempstead, Herts. HP2 4LT

ADVERTISERS INDEX

ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
AI International	OOP Languages & Environments	500	87
Applied Logic	Object Oriented Programming	498	84
Atom Style	Hardware	495	82
Biton	Clipper Software	506	93
Bits & Pieces	Peripheral/Expansion Products	512	91
Bits Per Second I	Graphics for Visual Basic	459	35
Bits Per Second II	Graphics for Clipper	460	37
Blink	Clipper Fast Linker	451	26
BL Security	Software Protection	504	92
Borland	Borland C++	429	7
Brent Communications	MAX copy Protection	447	21
CEBRA Communications	Multi VGA Adapters	471	49
Clearsoft	Software Security	463	41
CTL	Copy Protection Hardware	499	84
Cocking & Drury	Smalltalk V	476	53
DES	Software Protection	492	79
Digital	DR DOS	501	88
Evergreen	Case Tools	490	76
Expert Systems	Prolog 2 for Windows 3.0	511	91
FACTS Direct	FAX Service	502	90
G-Force	Graphical Interface for Clipper	464	42
Glockenspiel	Class Constructors	509	OBC
GWI	Software Design Tools	484	67
Grey Matter	Programming Tools	427	3
GUI-Euro Ltd	GUI Software	466	44
Guildsoft	PC Automation Software	461	37
HS Systems	8051 C Compiler	503	91
Inst. Analysts/Programmers	Institute	470	49
Instruction Set	Training	444	18
Instrumatic	Oregon C++	478	59
Intasoft	Software Management System	493	79
ITEL	Applications Generator	497	83
JPI	C++ Compilers	474	51
Lahey	Fortran Compiler	472	51
LPA	AI/KB/OOPS Software	452	29
Magnifeye	Software Protection Device	489	75
Microft	Security Software	507	94

ADVERTISER	PRODUCT/SERVICE	CIRCLE	PAGE
Microsoft Press	Book Publishers	481	65
Nantucket	Clipper	449/450	23/25
Next	Workstation	433	16
Nu-Mega	Debugging Tools	430	8
PC Bookshop	Book Specialists	448	21
Polyhedron	FORTRAN Compiler	483	67
Positive	Editor	462	38
QA	C Training	426	IFC
QBS I	Clipper Function Library	457	32
QBS II	Programmers Editor	482	66
QBS III	Clipper Add-Ons	488	73
Readmar I	Configuration Management	446	20
Readmar II	Tools	505	92
Real Techniques & Methods	CASE for Windows	478	5
Ridgeway	Computer Manuals/Binders	453	29
Roundhill	Development Tools	480	62
Salford S/W Marketing	FORTRAN for DOS & UNIX	485	67
Softlok International	Piracy Protection	508	97
Software Con. Co. I	Development Tools	458	35
Software Con. Co. II	Development Tools	477	57
Software Generation	Version Control	475	52
Software Paradise	Business Software	479	60/61
Software Security	Security Products	445	19
Solution Systems	Programming Editors	491	77
System C	Application Generator	487	70
System Science I	Maths Software	513	59
System Science II	Development Tools	486	69
System Star I	DBMS	432	13
System Star II	Software Tools	454	29
System Star III	Chart Drawing Software	456	31
System Star IV	DOS Extender	467	45
System Star V	Interface Management System	469	49
Thripplewoods	Sanyo Special Offer	510	85
Unisystems Software	Systems Management Tools	431	12
USA Software	Programming Tools	468	47
User Friendly	Software Copy Protection	455	31
Xoren	File Transfer Software	465	43

STOB - Not fairies' footfalls

'Backing up? Why, backing up is for punters!' (Traditional).

Finish typing in ammended comment and hit Alt-V for **V**erity's Compile, Link, Save to disk and Run (it only took me about eight hours in macro programming to set this up, which quite justifies the two minutes it must save me each week). The server is right behind me, so I can hear the familiar rhythm of the 300 MB hard disk as the compiler shifts up through the gears, you know, I bet I could identify most of the software on our network blindfolded, I'm so familiar with the different sounds...

Hold up. Put smallthought on Pause a sec. It's *not* making the usual noise. Instead of fairies' muffled footfalls on drum, we have distant scream like inexpertly wielded bluntish drill bit skidding on car wing. Listen. There it is again.

Now, Verity, don't panic. Ignore cold feeling on nape of neck, suppress urge to hiss 'If you have lost this file I will *bite* you' into server's A: drive. Be calm. Be rational. Ctrl-C. There; that's broken out of the macro. Attagirl. **F**ile. Save **A**s. `C:\PLEASE.CPP`. And, with careful finger, press `ENTER` key, so that, in a second, in a second, 'QEMM has detected an attempt to execute an illegal instruction in the program you were running. Press E to end the program, or R to reboot the computer.' Or how about 'B' to tell QEMM to **B**og off and mind its own **b**loody **b**usiness.

Well, that's three hours work down the drain, ha-ha (no, that was a light laugh, no, not at *all* like that woman in *Casualty* whose head exploded), so I suppose we may as well find out what's up with server. DIR C:\, two trillion things flick by at the speed they do on a 486, fine. DIR D:\, same thing. DIR E:\, someone switches on the Black and Decker again, we have a General Failure Reading Drive E:\, guess which drive someone has entrusted with all her recent work, General Failure Verity's Temper; Abort, Retry, Ignore, Scream.

Look at backup log, to discover who made most recent weekly copy of server. Dave. Uh-huh. You remember Dave - the man with two talents: he can solder an IC chip holder onto a bit of vero while squinting through the smoke of a Silk Cut Menthol, and he cheats on his wife every Thursday night - she believes it has taken a year to repaint the TA hut. Dave's just the chap to be entrusted with my life's, ok week's work.

Oh, Hi Dave, you couldn't just lend me the tape of the backup that you made last Friday. What's that? You didn't do a *full* backup? So what sort of backup did you make, you old cockchafer? You made a partial backup of C: and D: did you, you old cockchafer. Well you know what, you old cockchafer, you're not fit to back up a 1969 Mini on a disused airstrip, never mind a computer, you old...

And so, as we leave Verity swearing at her colleague, Betty the Backup Bunnie says: 'Remember, Readers, a few minutes spent backing up your work is more than just a few minutes lost drinking time.'

This edition of Stob was sponsored by the HM Government Department for Getting People to do Sensible Things for Their Own Good. For a free leaflet, please write to the Abandoned Fridge, Unreported Gas Leak Road, East Condom, Cholesterolshire.

.EXE

WANTED

to

Develop high performance networking products:

- **Software Engineers**
- **Hardware Engineers**

Madge Networks, the world leader in token ring networks for PCs, needs more excellent engineers. Our engineers:

- Develop World Class Products
- Have fun in the process
- Form a highly-motivated team
- Make a major contribution to a company doubling in size each year.

Personal excellence is an absolute requirement. All applicants will have at least one of:

- A First Class Honours Degree
- 3 A grades in numerate A-level subjects
- A world class flair for developing computer systems

Previous experience of networking is not required. Starting salaries for engineers range up to £30K, depending on experience.

To apply, please send a full CV to Fiona Woods at:

Madge Networks Ltd
100 Lodge Lane
Chalfont St. Giles
Bucks HP8 4AH